



UNIVERSIDAD
POLITECNICA
DE VALENCIA



Un Método de Evaluación de Usabilidad de Mashups Basado en la Composicionalidad de sus Componentes

Tesis de Máster en Ingeniería del Software,
Métodos Formales y Sistemas de Información
(MISMFSI)

Grupo de Ingeniería del Software y Sistemas de Información (ISSI)
Departamento de Sistemas Informáticos y Computación (DSIC)
Universidad Politécnica de Valencia (UPV)

2013

Irene Priscila Cedillo Orellana

Directores:

Dr. Emilio Insfrán Pelozo

Dr. Adrián Fernández Martínez

Un Método de Evaluación de Usabilidad de
Mashups Basado en la Composicionalidad
de sus Componentes.

Irene Priscila Cedillo Orellana

Impreso en Valencia, España

2013

AGRADECIMIENTOS

A mis directores de tesis Emilio Insfrán y Adrián Fernández por todo su apoyo, consejos, enseñanzas y ayuda en la elaboración de este trabajo.

A mi País por darme la oportunidad de prepararme académicamente y permitir mi realización profesional brindándome los medios necesarios para cumplir este propósito.

A mi esposo querido Freddy por ser mi fuerza y mi apoyo, la inspiración que necesito.

A mi hija Juliana por ser la razón de mi vida y darme esa bella sonrisa cada vez que regreso a casa, recordándome que debo ser siempre mejor.

A mis padres, por todo lo que diariamente han hecho y siguen haciendo por mí, son los mejores.

A mis compañeros de laboratorio, que hacen que el trabajo diario se convierta en una alegría.

Contenido

RESUMEN	9
ABSTRACT	11
RESUM	12
Capítulo 1. Introducción	13
1.1. Motivación	13
1.2. Objetivos	14
1.3. Tareas de Investigación	14
1.4. Estructura del Trabajo	15
Capítulo 2. Base Tecnológica	17
2.1. Tecnologías Web	17
2.1.1. La Web 2.0	17
2.1.2. Aplicaciones Web	17
2.1.3. Arquitectura Orientada a Servicios (SOA)	17
2.1.4. Java Servlet	18
2.1.5. Portlets	18
2.1.6. End-User Development (EUD)	18
2.1.7. Mashups	19
2.2. Estándares de Calidad Dirigidos a la Usabilidad	25
2.2.1. Estándares Orientados al Proceso: ISO 9241 e ISO 13407	26
2.2.2. Estándares Orientados al Producto: ISO 9126 e ISO 14598	28
2.2.3. Estándar ISO 25000 (SQuaRE)	30
2.3. Modelo de Referencia para la Evaluación de los Productos de Software	32
2.3.1. Entradas de la Evaluación	32
2.3.2. Salidas de la Evaluación	33
2.3.3. Restricciones para la Evaluación.	34
2.3.4. Recursos para la evaluación.	35
Capítulo 3. Estado Actual de la Investigación	41
3.1. Introducción a los Estudios de Mapeos Sistemáticos	41
3.2. Mapeo Sistemático sobre la Calidad de Producto en Mashups	42
3.2.1. Fase de Planificación	42
3.2.2. Fase de Conducción	47

3.2.3.	Reporte de resultados	48
3.3.	Mapeo Sistemático sobre la Composicionalidad de los Mashups	53
3.3.1.	Fase de Planificación	54
3.3.2.	Fase de Conducción.....	58
3.3.3.	Reporte de Resultados	59
Capítulo 4.	Modelo de Usabilidad para Mashups.....	65
4.1.	Facilidad de Entendimiento (Appropriateness recognisability)	66
4.2.	Facilidad de Aprendizaje (Learnability)	71
4.3.	Operatividad.....	74
4.4.	Protección de Errores al Usuario.....	78
4.5.	Estética de Interfaces de Usuario.....	79
4.6.	Accesibilidad.....	80
4.7.	Conformidad.....	80
Capítulo 5.	Método de Evaluación de la Usabilidad de Mashups	82
5.1.	Definición del proceso con SPEM 2.....	83
5.2.	Método de Evaluación con SPEM 2.....	86
5.2.1.	Especificación de los Requisitos de Evaluación.....	89
5.2.2.	Especificación de la Evaluación	91
5.2.3.	Diseño de la Evaluación.....	95
5.2.4.	Ejecución de la Evaluación e Informe de Usabilidad.....	97
5.2.5.	Finalización de la Evaluación	99
Capítulo 6.	Aplicación de Método de Evaluación	100
6.1.	Mashups a ser Evaluados	100
6.1.1.	Liveplasma	100
6.1.2.	Caribbean Resorts Mashups.....	101
6.2.	Aplicación del Proceso de Evaluación de Usabilidad	106
6.2.1.	Evaluación Liveplasma.....	106
6.2.2.	Evaluación del Mashup Caribbean Resorts	121
6.2.3.	Evaluación Skypicker	128
6.3.	Conclusiones de la Aplicación del Método de Evaluación	135
Capítulo 7.	Conclusiones y Trabajos Futuros.....	136
7.1.	Conclusiones.....	136

7.2. Trabajos Futuros.....	138
7.3. Publicaciones relacionadas	138
REFERENCIAS.....	140
APENDICE A – BIBLIOGRAFIA MAPEO SISTEMATICO CALIDAD	144
APENDICE B – BIBLIOGRAFIA MAPEO SISTEMATICO COMPOSICIONALIDAD	148
APENDICE C: PLANTILLAS	157
APENDICE D: PLANTILLAS DE CASOS DE ESTUDIO	158

INDICE DE FIGURAS

Figura 1-1	Metodología de investigación utilizada	15
Figura 1-2	Estructura del trabajo.....	16
Figura 2-1	Actividades en el diseño del método de evaluación.....	27
Figura 2-2	Calidad interna y externa.	28
Figura 2-3	Calidad en uso Modelada.....	28
Figura 2-4	Calidad Proceso, Producto y En Uso.....	29
Figura 2-5	Organización de las series SQuaRE.....	31
Figura 2-6	Modelo de medición de la calidad del producto software según SQuaRE.	32
Figura 2-7	Proceso de evaluación para productos de software.....	32
Figura 2-8	Entradas de la evaluación de los Mashups	33
Figura 2-9	Salidas para la evaluación	34
Figura 2-10	Restricciones para la evaluación	35
Figura 2-11	Recursos para la evaluación	36
Figura 2-12	Calidad en el ciclo de vida de un producto de software	38
Figura 2-13	Características calidad en uso	38
Figura 2-14	Características de las medidas internas de calidad.....	39
Figura 2-15	Características de las medidas externas de calidad	39
Figura 3-1	Pasos de un Mapeo Sistemático	42
Figura 3-2	Publicaciones sobre Calidad en Mashups.	50
Figura 3-3	Comparación entre criterios de extracción C1, C2, C5.....	50
Figura 3-4	Criterios de extracción Artefactos-Características de Calidad de Producto--Tipos de enfoque	51
Figura 3-5	Criterios de extracción Artefactos-Características de Calidad de Producto—Estados del ciclo de vida-Validación	52
Figura 3-6	Relaciones entre criterios de extracción. Mapeo Sistemático Composicionalidad	62
Figura 3-7	Relaciones entre criterios de extracción. Mapeo Sistemático Composicionalidad	63
Figura 5-1	Metamodelo SPEM.....	84
Figura 5-2	Mapeo contenido-proceso SPEM 2.....	84
Figura 5-3	Proceso de construcción de un componente.....	86
Figura 5-4	Diagrama del ciclo de vida de construcción de un Mashup con SPEM.....	87

Figura 5-5	Pasos del Proceso de Evaluación de la Usabilidad de Mashups.....	88
Figura 5-6	Proceso de evaluación de usabilidad para Mashups	89
Figura 5-7	Especificación de los “Requisitos de Evaluación”	90
Figura 5-8	Especificación de la evaluación	92
Figura 5-9	Diseño de la evaluación.....	96
Figura 5-10	Fase “Ejecución de la evaluación”	97
Figura 5-11	“Finalizar la evaluación”	99
Figura 6-1	Mashup Liveplasma www.liveplasma.com	100
Figura 6-2	Pantalla principal Liveplasma	101
Figura 6-3	Mashup Caribbean Resorts http://www.gooseflap.net/	102
Figura 6-4	Componente del Mashup Caribbean Resorts	102
Figura 6-5	Componente del Mashup Caribbean Resorts	103
Figura 6-6	Componente de YouTube del Mashup Caribbean Resorts Skypicker	103
Figura 6-7	Mashup Skypicker www.skypicker.com	104
Figura 6-8	Parte superior Pantalla Skypicker.....	104
Figura 6-9	Pantalla Skypicker vista de componentes principales.....	105
Figura 6-10	Pantalla “leave a message” Skypicker	105
Figura 6-11	Pantalla de pago Skypicker.....	106
Figura 6-12	Disposición de componentes en Liveplasma	107
Figura 6-13	Lista de componentes Caribbean Resorts Mashup.....	121
Figura 6-14	Métricas a ser evaluadas en el caso de estudio Caribbean Resorts.....	123
Figura 6-15	Instantánea Mashup Caribbean Resorts.	125
Figura 6-16	Disposición de componentes en Skypicker	129

INDICE DE TABLAS

Tabla 2-1	Ejemplos de herramientas existentes para la construcción de Mashups.	25
Tabla 2-2	Estándares ISO 92941.....	26
Tabla 3-2	Mapeo Sistemático. Resultados según criterios de extracción.....	48
Tabla 3-4	Criterios de extracción. Mapeo sistemático sobre composicionalidad	57
Tabla 3-5	Resultados fase de conducción	59
Tabla 4-1	Facilidad de entendimiento.....	71
Tabla 4-2	Facilidad de aprendizaje.....	73
Tabla 4-3	Atributos de la sub-característica “Operatividad”	78
Tabla 4-4	Sub-característica Estética de interfaces de usuario.....	80
Tabla 4-5	Sub-característica Accesibilidad.	80
Tabla 5-1	Nomenclatura SPEM 2.....	85
Tabla 6-1	Atributos a evaluar en el Mashup Liveplasma	108
Tabla 6-2	Métricas a ser evaluadas en el caso de estudio Liveplasma	109
Tabla 6-3	Problema de usabilidad 001 detectado.	112
Tabla 6-4	Problema de usabilidad 002 detectado.	112
Tabla 6-5	Problema de usabilidad 003 detectado	114
Tabla 6-6	Problema de usabilidad 004 detectado.	114
Tabla 6-7	Problema de usabilidad 005 detectado	115
Tabla 6-8	Problema de usabilidad 006 detectado	116
Tabla 6-9	Problema de usabilidad 007 detectado	117
Tabla 6-10	Problema de usabilidad 008 detectado	118
Tabla 6-11	Problema de usabilidad 009 detectado	119
Tabla 6-12	Problema de usabilidad 010 detectado	120
Tabla 6-13	Problema de usabilidad 001 detectado	126
Tabla 6-14	Problema de usabilidad 006 detectado	128
Tabla 6-15	Definición de Métricas de Calidad.....	131
Tabla 6-16	Problema de usabilidad 001 detectado	133
Tabla 6-17	Problema de usabilidad 002 detectado.	134
Tabla 6-18	Problema de usabilidad 003 detectado	135

RESUMEN

Los Mashups son una nueva generación de aplicaciones Web que integran componentes provistos por terceros desde la Web. La usabilidad en este tipo de aplicaciones es un factor muy importante ya que como en toda aplicación Web, este determina el éxito de la aplicación. Si bien se han planteado métodos de evaluación que permiten evaluar la usabilidad de aplicaciones Web en general, éstos no cubren las características inherentes a la composicionalidad de los Mashups.

La presente tesina de máster pretende aportar un método de evaluación de la usabilidad de Mashups acorde con sus características propias de composicionalidad. El método está compuesto de un modelo de usabilidad y un proceso de evaluación que provee guías sobre la forma en la que el modelo de usabilidad puede ser usado para realizar evaluaciones específicas. Tanto el modelo de usabilidad como el proceso de evaluación están alineados con el último estándar ISO/IEC 25000 para la evaluación de la calidad de producto (SQuaRE) y ambos toman en cuenta la naturaleza composicional de los Mashups. El modelo de usabilidad descompone el concepto de usabilidad en sub-características y métricas genéricas. Estas métricas son aplicadas en la composición del producto final y el método de evaluación puede ser aplicado en cualquier estado del ciclo de vida de este tipo de aplicaciones, ya sea en la selección de los componentes, durante el proceso de composición o cuando el producto está listo para ser usado.

Para apoyar esta propuesta se ha realizado un profundo estudio del estado del arte. Este estudio comprende dos mapeos sistemáticos: el primero abarca un estudio de la evaluación de la calidad de los Mashups y el segundo cubre las características de la composicionalidad envueltas en los Mashups. Los resultados obtenidos aportaron de una forma significativa en la definición del Modelo de Usabilidad de Mashups y posteriormente en el proceso de evaluación de usabilidad.

Finalmente, se desarrollaron tres casos de estudio que muestran la viabilidad de nuestro enfoque. Estos casos de estudio muestran en detalle como el método de evaluación propuesto utiliza nuestro Modelo de Usabilidad para Mashups. Los resultados muestran que nuestra propuesta permite detectar problemas de usabilidad, los cuales una vez corregidos, permiten obtener Mashups más usables.

ABSTRACT

Mashups are the new-generation of Web applications that integrates third-party components from the Web. Usability in these applications is very important since their success depends on this quality factor. Although several methods have been proposed to evaluate the usability of Web applications in general, they do not cover the specific characteristics inherent to the compositionality of Mashups.

This master thesis aims to provide a method for evaluating the usability of Mashups according to their specific characteristics of compositionality. The method is composed of a usability model and an evaluation process that provides guidelines on how the usability model can be used to perform specific evaluations. Both the usability model and the evaluation process are aligned with the latest ISO/IEC 25000 standard for software product quality evaluation (SQuaRE), and both take into account the compositional nature of mashups. The usability model breaks down the concept of usability into sub-characteristics, attributes and generic metrics. These metrics are applied to the final product composition and process evaluation method can be applied at any stage of the life cycle of this kind of applications (i.e., during the component selection, the composition process or when the product is ready for use).

In order to support the foundation of this proposal, an in-depth study was performed about the state of art. This study comprises two systematic mappings: the first one covers the quality evaluation of Mashups, whereas the second one covers the compositionality features involved in Mashups. The obtained results were used as input in order to define both our Mashups Usability Model, and our usability evaluation process.

Finally, we developed three case studies in order to show the feasibility of our approach. These case studies show in detail how the proposed evaluation method was followed by using our Mashup Usability Model. Results showed that our approach was able to detect usability problems, which once corrected, it allows obtaining more usable Mashups.

RESUM

Els Mashups són una nova generació d'aplicacions Web que integren components proveïts per tercers des de la Web. La usabilitat en este tipus d'aplicacions és un factor molt important ja que com en tota aplicació Web, este determina l'èxit de l'aplicació. Si bé s'han plantejat mètodes d'avaluació que permeten avaluar la usabilitat d'aplicacions Web en general, estos no cobreixen les característiques inherents a la composicionalitat dels Mashups.

La present tesina de màster pretén aportar un mètode d'avaluació de la usabilitat de Mashups d'acord amb les seues característiques pròpies de composicionalitat. El mètode està compost d'un model d'usabilitat i un procés d'avaluació que proveïx guies sobre la forma en què el model d'usabilitat pot ser usat per a realitzar avaluacions específiques. Tant el model d'usabilitat com el procés d'avaluació estan alineats amb l'últim estàndard ISO/IEC 25000 per a l'avaluació de la qualitat de producte (SQuaRE) i ambdós prenen en compte la naturalesa composicional dels Mashups. El model d'usabilitat descompon el concepte d'usabilitat en sub-característiques i mètriques genèriques. Estes mètriques són aplicades en la composició del producte final i el mètode d'avaluació pot ser aplicat en qualsevol estat del cicle de vida d'este tipus d'aplicacions, ja siga en la selecció dels components, durant el procés de composició o quan el producte està llest per ser usat.

Per a recolzar esta proposta s'ha realitzat un profund estudi de l'estat de l'art. Este estudi comprèn dos mapeos sistemàtics: el primer comprèn un estudi de l'avaluació de la qualitat dels Mashups i el segon cobreix les característiques de la composicionalitat en els Mashups. Els resultats obtinguts varen aportar d'una forma significativa en la definició del Model d'Usabilitat de Mashups i posteriorment en el procés d'avaluació d'usabilitat.

Finalment, es van fer tres casos d'estudi que mostren la viabilitat del nostre enfocament. Estos casos d'estudi mostren en detall com el mètode d'avaluació proposat utilitza el nostre Model d'Usabilitat per a Mashups. Els resultats mostren que la nostra proposta permet detectar problemes d'usabilitat, els quals una vegada corregits, permeten obtindre Mashups més usables.

Capítulo 1. Introducción

1.1. Motivación

Con la Web 2.0 se ha iniciado una era de interacción Web en la cual se hacen necesarias nuevas herramientas que permitan combinar contenidos y servicios en aplicaciones compuestas denominadas Mashups. Incontables actividades tales como efectuar compras on-line, realizar transacciones, ubicar lugares, entre otras han hecho necesario que el usuario interactúe de una manera directa en la construcción de aplicaciones para su propio consumo. Estas, se han convertido en un nuevo reto de investigación tanto para la academia como para la industria.

Los ingenieros del software y los programadores, en muchas ocasiones, no conocen tan claramente el proceso de negocio como el usuario que requiere una aplicación puntual, haciendo necesaria la intervención directa del cliente final en la construcción del Mashup.

Las empresas necesitan herramientas que serán utilizadas por determinado lapso de tiempo, éstas no forman parte de sus sistemas de información habituales, sino más bien son construidas para satisfacer una necesidad “situacional” de la empresa. El comercio electrónico se ha visto en un ascenso vertiginoso, por lo cual se han desarrollado muchos sistemas que proveen comparativas de precios, como es el caso de los intermediarios en venta de planes de viajes, que necesitan datos y componentes proporcionados por terceros para que el cliente pueda escoger la opción que más se ajuste a sus requisitos; los sitios de bienes raíces tales como Housingmaps.com el mismo que toma la lista de casas, apartamentos y habitaciones que están de venta o alquiler desde Craigslist y los muestra sobre un mapa de Google.

Todas estas necesidades se enmarcan perfectamente para ser solventadas con los Mashups.

Las aplicaciones Web Mashups son desarrolladas por medio de la composición de contenidos o datos, lógica de aplicación e interfaces de usuario desde varias fuentes en la red. La principal diferencia con las aplicaciones tradicionales basadas en componentes es que los Mashups sirven para una situación muy específica y generalmente por un período corto de tiempo [68].

Junto con el advenimiento de este nuevo concepto, se han presentado nuevos retos para hacer que el proceso de composición y desarrollo sea sencillo para el usuario final, que generalmente no posee conocimientos de programación, para esto se han implementado una serie de herramientas automáticas y semi-automáticas que permiten la construcción asistida de este tipo de aplicaciones. En algunos trabajos se han estudiado formas de optimizar la calidad de los componentes [12], de mejorar los aspectos de seguridad del producto final [25][44][56][59], entre otras características no funcionales, sin embargo poco se ha hablado sobre un modelo de calidad en cuanto a la usabilidad del producto terminado.

Así también existen modelos de calidad en cuanto a la usabilidad que están dirigidos a aplicaciones Web en general [20], pero no tratan aspectos específicos que, dada la naturaleza de los Mashups se requieren para obtener productos con alta calidad en lo que a usabilidad se refiere.

Los componentes utilizados en la elaboración de este tipo de aplicaciones Web, no solamente involucran artefactos de la capa de interface sino también integran datos y procesos de negocio, siendo necesario un estudio que muestre la injerencia de estos componentes en la usabilidad del producto final.

Por tanto se hace necesario el contar con método que permita evaluar la usabilidad de los Mashups basado en la composicionalidad de sus componentes.

1.2. Objetivos

Este trabajo tiene como objetivo general, definir un método que permita evaluar la usabilidad de una aplicación Web Mashup basada en la composicionalidad de sus componentes.

Adicionalmente, se va a realizar un estudio que recoja el estado actual sobre aspectos de composicionalidad y calidad de los Mashups, a fin de direccionar de manera objetiva la investigación que se presentará en este trabajo.

Como objetivos específicos que ayudarán a conseguir lo anteriormente expuesto tenemos:

- Revisar las tecnologías asociadas a la Web 2.0 y que intervengan en una manera directa con los Mashups.
- Realizar un mapeo sistemático sobre el estado actual de los Mashups, que permita conocer los aspectos de composicionalidad y calidad de este tipo de productos de software.
- Analizar los modelos de calidad existentes y que puedan servir de base para la elaboración de un modelo específico para la usabilidad de los Mashups.
- Crear un modelo de usabilidad de Mashups, que permita al evaluador y por qué no al compositor a evaluar y crear respectivamente a fin de conseguir productos altamente usables.
- Definir un método de evaluación de la usabilidad de los Mashups que utilice el modelo de calidad propuesto.
- Ejemplificar por medio de casos de estudio la metodología planteada.

Al existir varias herramientas que faciliten la construcción del Mashup y específicamente guíen la composición para obtener productos de una manera más fácil desde la perspectiva de la calidad del proceso, este trabajo cubrirá específicamente la calidad del producto final obtenido luego del proceso de la composición y que será desarrollado ya sea por un usuario experto en el dominio sin conocimientos de programación o por programadores que deseen realizar una integración rápida y que hagan uso de componentes provistos por terceros, para una situación concreta ya sea dentro de la empresa, negocio o consumo propio.

1.3. Tareas de Investigación

El método de investigación a seguir involucra una serie de pasos que nos permitirán obtener los resultados esperados. A continuación se detalla cada uno de ellos:

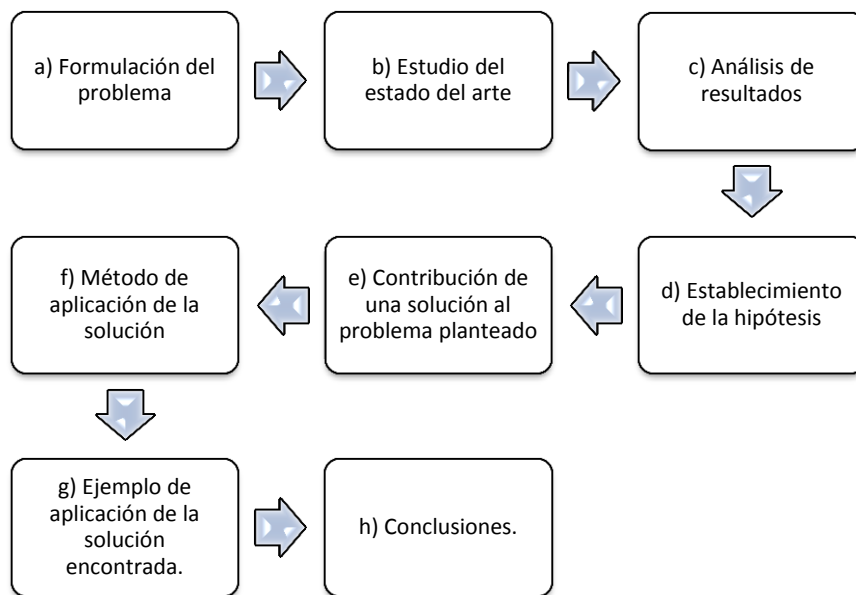


Figura 1-1 Metodología de investigación utilizada

- a) **Formulación del problema:** Se realizará una delimitación del mismo para poder tener un punto central y claro al cual atacar en el proceso de la búsqueda de la solución.
- b) **Estudio del estado del arte:** Para el estado del arte se han realizado dos estudios consistentes en mapeos sistemáticos, que permitirán tener un estado actual de los aspectos de calidad y composicionalidad referentes a los Mashups.
- c) **Análisis de resultados:** A continuación se analizaron los resultados de los estudios del estado del arte y se delimitaron los aspectos que se necesitan investigar en el presente trabajo de acuerdo a la literatura científica actual.
- d) **Método de aplicación de la solución:** Se encontró que existe una falta de estudios referentes a la calidad en la usabilidad de los Mashups y en base a esto se estableció la hipótesis que se requiere esclarecer en la presente investigación.
- e) **Contribución de una solución al problema planteado:** Se elaboró un modelo de usabilidad que sirva de base a la evaluación de la calidad de usabilidad para Mashups.
- f) **Método de aplicación de la solución:** Se elaboró una metodología para la evaluación de la usabilidad de los Mashups, en la cual se tomará como entrada el modelo de usabilidad propuesto y se darán pautas para la mejora de la calidad en usabilidad de este tipo de aplicaciones.
- g) **Ejemplo de aplicación de la solución encontrada:** En este paso se aplicará el método de evaluación de usabilidad a cuatro Mashups, cubriendo los principales atributos de usabilidad dirigidos a este tipo de aplicaciones Web.
- h) **Conclusiones:** Se sacarán las principales conclusiones relacionadas al método de evaluación de los Mashups.

1.4. Estructura del Trabajo

Este trabajo está compuesto por 7 capítulos distribuidos de la siguiente manera:

El **capítulo 1** introduce al lector en el contexto y en el alcance del trabajo, indicará la metodología y la organización del mismo.

El **capítulo 2** recoge todos los conceptos necesarios para el desarrollo de la tesina, la taxonomía útil para el desarrollo del estado del arte y demás temas relacionados con el problema a resolver.

El **capítulo 3** presenta el estado actual, por medio de un mapeo sistemático y las cifras de las clasificaciones en las cuales se centra el estudio y que permitirán tener una vista objetiva del problema que se resolverá.

Los **capítulos 4 y 5** muestran la contribución de este trabajo. El **capítulo 4** contiene el modelo de usabilidad de Mashups compuesto por un conjunto de características, sub-características, atributos de calidad y métricas para medir los atributos. El modelo de usabilidad tiene en cuenta el producto final resultante del proceso de composición del Mashup. El **capítulo 5**, muestra cómo realizar la evaluación de la usabilidad del Mashup, mediante la aplicación del modelo propuesto.

En el **capítulo 6** se trata un caso de estudio en el cual se muestra cómo realizar la evaluación Mashup aplicando el modelo propuesto.

El **capítulo 7** contiene las conclusiones del trabajo, las publicaciones generadas y los trabajos futuros.

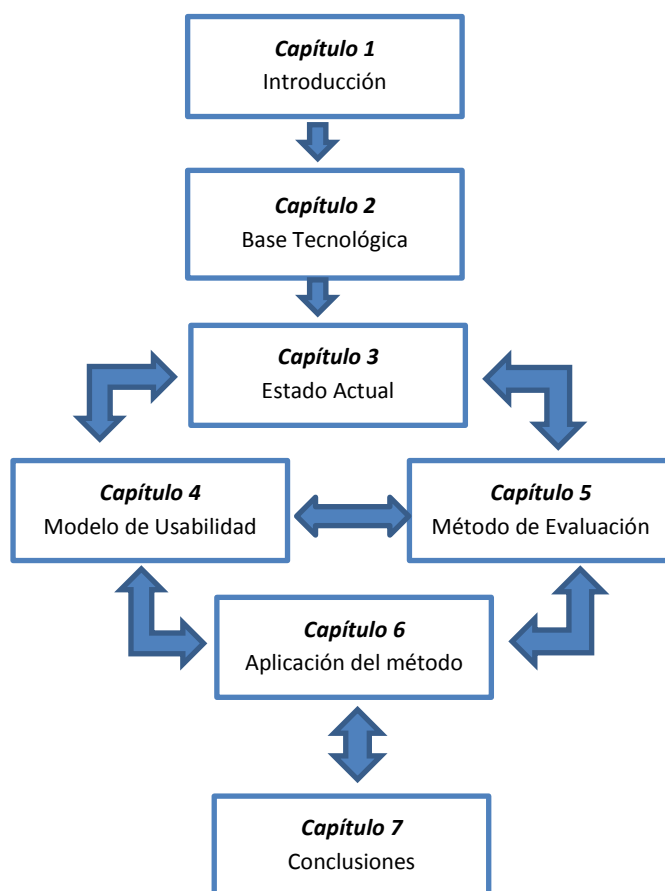


Figura 1-2 Estructura del trabajo

Capítulo 2. Base Tecnológica

2.1. Tecnologías Web

2.1.1. La Web 2.0

La Internet y las tecnologías relacionadas han creado un mundo interconectado, en el cual nosotros podemos intercambiar información fácilmente, ejecutar tareas de procesos colaborativo y comunicarnos entre usuarios con intereses similares [67].

La Web 2.0 ha iniciado una nueva era de interacción Web. Incontables actividades todos los días tales como buscar información, comprar, llenar formularios y planificar citas, han sido hechas de manera efectiva y generalmente menos costosa en la Web. Para quien la Web es inaccesible por cualquier razón puede volverse incrementalmente excluyente del estilo de vida común sino logra acceder a ella [69].

2.1.2. Aplicaciones Web

En la ingeniería de software se denomina aplicación Web a una herramienta informática, que un usuario puede utilizar a través de internet o de una intranet y que puede ser accedida a través de un navegador.

Las aplicaciones Web se han popularizado debido a lo fácil que es acceder a ellas por medio del navegador como cliente ligero, a la independencia del sistema operativo y a la gran facilidad en mantenimiento que presentan.

2.1.3. Arquitectura Orientada a Servicios (SOA)

La arquitectura orientada a servicios (SOA), es una arquitectura de software que permite la definición de la utilización de servicios para dar soporte a los requisitos del negocio.

Con la introducción del incremento de la complejidad de los servicios Web en los últimos tiempos, ha habido una explosión del interés en este tipo de arquitectura. Un estilo estructural cuya meta es conseguir un acoplamiento de servicios que interactúan habiendo mucho estudio en el tema [6].

Las arquitecturas orientadas a servicio (SOA) están bien establecidas como un paradigma arquitectural para sistemas distribuidos. Durante los últimos años SOA ha pasado de ser un concepto promovido por unos pocos para convertirse en uno de los estilos arquitectónicos más importantes de las empresas, que están tratando de adoptarlas.

SOA permite la creación de sistemas de información altamente escalables que reflejan el negocio de la organización, a su vez brinda una forma bien definida de exposición e invocación de servicios (comúnmente pero no exclusivamente servicios Web), lo cual facilita la interacción entre diferentes sistemas propios o de terceros.

Según la definición del equipo de trabajo de Open Group SOA, la arquitectura SOA es estilo arquitectural que soporta orientación a servicios.

La orientación a servicios es un modo de pensar en término de servicios y desarrollo basado en servicios.

Un servicio es una representación lógica de una actividad repetible de procesos de negocios que tiene una salida especificada, puede ser compuesta por otros servicios y es una caja negra para consumidores de los servicios.

2.1.4. Java Servlet

Los servlets son objetos que residen en un contenedor de servlets como lo es por ejemplo Tomcat. La palabra servlet viene de applet que consistían en pequeños programas que se ejecutan en el contexto de un navegador Web.

El uso más común de un servlet es generar páginas Web dinámicas que sean ejecutadas en un servidor Web.

2.1.5. Portlets

Los portlets son componentes modulares de interfaces de usuario que se encuentran en un portal Web. Estos son contruidos a través de código de marcado que se agrega en el portal. Los portales pueden estar contruidos por tanto de una colección de ventanas que no se solapan, y cada una de ellas muestra un contenido perteneciente al portlet en mención.

Además, los portlets proporcionan un objeto que almacena las preferencias de usuario, este se llama PortletPreferences. Estas preferencias se almacenan en una base de datos persistente, de este modo se encuentran disponibles cada vez que el contenedor de portlets se inicie. El desarrollador no necesita implementar el almacenamiento.

Los portlets al igual que los servlets son manejados por un contenedor especializado como lo es Tomcat por ejemplo, generando contenido dinámicamente.

2.1.6. End-User Development (EUD)

El EUD es un tópico de investigación en el campo de las ciencias de la computación y la interacción humano-máquina, que describe actividades o técnicas que permiten a los usuarios finales programar ordenadores. La gente quien no es un desarrollador profesional, puede usar herramientas de EUD para crear o modificar artefactos de software (descripciones de comportamiento automático) y objetos de datos complejos sin necesidad de un conocimiento significativo de un lenguaje de programación.

En [17] se propone la siguiente definición:

“El Desarrollo dirigido al usuario final, puede ser definido como un conjunto de métodos, técnicas y herramientas que permite a los usuarios de sistemas de software, quienes no son desarrolladores profesionales de software, en algún momento crear, modificar o extender un artefacto de software”.

Por tanto, EUD permite a los usuarios finales diseñar o personalizar la interface y funcionalidad de software. Esto es útil porque los usuarios conocen sus propias necesidades mejor que un programador y ellos pueden a menudo aportar con mejoras cuando crean necesario.

2.1.7. Mashups

El término Mashup tiene su origen en el dominio musical, refiriéndose a artistas que mezclan algunas piezas de música, usualmente desde diferentes estilos musicales, en un simple registro. Este término ha ido generalizándose y abarcando otros dominios que introducen la idea de derivar un nuevo trabajo mezclando dos o más piezas.

Los Mashups en el ámbito de la Informática, son aplicaciones basadas en la Web que son creadas por la combinación y procesamiento de recursos provistos por terceros, que contribuyen con datos, presentación o funcionalidad [38].

La principal razón por la cual los Mashups han ganado popularidad es que no necesitan que gente con conocimientos técnicos los desarrollen, pudiendo hacerlo gente sin destrezas de programación pero con claro conocimiento del dominio del problema.

La Ingeniería de los Mashups puede ser considerada como una especialización de la ingeniería Web, la cual es una especialización a su vez de la Ingeniería del software. En contraste con la mayoría de programas en computación, los Mashups no están en la línea de la programación y son creados gráficamente. Los Mashups tienen un amplio rango de usuarios, incluyendo expertos no programadores, en el contexto de los Mashups, los usuarios finales juegan un rol importante ya que no solamente se centran en el uso de la aplicación sino además participan activamente en su evolución.

En particular, el proceso de desarrollo de los Mashups incorpora tres actores: El desarrollador de los componentes, quien es responsable de los detalles técnicos, desarrolla los componentes de los Mashups escribe la documentación y publica el componente en la red. El que compone el Mashup, crea la aplicación y el usuario quien usualmente es el “desarrollador” del mismo.

En definitiva, los Mashups son una nueva generación de aplicaciones que soportan la “composición” de aplicaciones partiendo de contenidos y servicios provistos por terceras partes y puestos a disposición en la Web. Los Mashups fueron inicialmente concebidos en el contexto de la Web de consumo como un modo de que los usuarios finales puedan crear sus propias aplicaciones. Más recientemente, los Mashups empresariales comenzaron a emerger, como instrumentos para los usuarios promedio para crear aplicaciones situacionales fácil y rápidamente ya sea por un corto período para soportar sus procesos de decisión. Los dos conceptos previos son caracterizados por el involucramiento de los usuarios finales. Recientemente algunos enfoques han sido propuestos para facilitar la tarea de composición. Por un lado algunas herramientas tratan de reducir el esfuerzo requerido para programar la composición de servicios, proponiendo abstracciones de alto nivel, además sustentadas en notaciones de facilidad de uso. Por otro lado algunos enfoques se centran en la producción de recomendaciones sobre los servicios de componentes y los patrones de composición a ser adoptados.

2.1.7.1.Arquitectura de los Mashups

La arquitectura de los Mashups está compuesta de tres partes:

- a) **El Proveedor de Contenidos**

Se refiere a la fuente de los datos. Los datos están disponibles vía una API, diferentes protocolos Web o artefactos disponibles en Internet, elaborados por terceras partes y que no han sido precisamente contruidos con un Mashup en particular como objetivo.

b) El Sitio Mashup

Es la nueva aplicación Web que provee un nuevo servicio utilizando diferente información de la que son dueños los proveedores.

c) El Web Browser Cliente

Es la interfaz de usuario del Mashup. En una aplicación Web, el contenido puede ser mezclado por los Web browser clientes usando lenguaje Web del lado del cliente o directamente presentando el Mashup desde el servidor a través del navegador.

2.1.7.2. Acceso a los Componentes

El acceso se refiere al lugar donde los componentes del Mashup son mezclados [62].

a) Mashups del Lado del Cliente

Se denominan Mashups del lado del cliente cuando su ejecución toma lugar en una plataforma de cliente tal como un browser.

Típicamente mantienen tiempos cortos de procesamiento y corren bajo el control directo del usuario.

Reciben y procesan contenidos “crudos” desde proveedores de contenidos en una localización externa y no controlada.

El Mashup libremente accede al contenido/servicios de las plataformas del proveedor para utilizar los contenidos y/o los servicios activos por lo cual no muestra una protección contra ataques.

b) Mashups del Lado del Servidor

Se denominan Mashups del lado del servidor a aquellos cuya ejecución toma lugar en una plataforma del lado del servidor.

Típicamente se caracterizan por un tiempo largo de procesamiento y no requieren el control directo de un usuario. Son permanentemente “demonios” activos para cuando ocurre una situación configurada.

La combinación toma lugar en un ambiente protegido. Los sistemas clientes solamente recibirán la combinación de contenidos que el Mashup decida proveer.

Los Mashups del lado del servidor protegen las plataformas por medio de una capa de procesamiento que inspecciona, autoriza o bloquea requisitos de contenido/servicios.

2.1.7.3. Fuentes de Integración de Mashups

La integración ha sido definida como un reto técnico significativo encontrado cuando desarrollamos Mashups. Los Mashups agregan varias fuentes de datos de diferentes tipos (ej. Bases de datos, sistemas legados, XML, páginas Web dinámicas, RSS, etc.) [9].

a) Integración Homogénea

La integración homogénea se centra en combinar datos similares desde diferentes fuentes, dos ejemplos podrían ser los Mashups Empresariales acoplados con sistemas legados y los Mashups de consumidores acoplados con servicios Web [9].

b) Integración Heterogénea

La integración heterogénea trata de como acoplar datos distintos desde diferentes fuentes, un ejemplo podría ser un Mashup Empresarial que combine sistemas legados y servicios Web públicos [9].

c) Integración de Aplicación

La integración de aplicación, se centra en acoplar distintos procesos de negocios en Mashups Empresariales a través de metodologías tales como SOA [9].

2.1.7.4. Formas de Descubrimiento de Componentes

Las formas de descubrimiento de componentes se refieren a la manera en la cual el desarrollador del Mashup puede integrar o realizar la tarea de composición de la aplicación.

Por ejemplo, existen numerosas herramientas que permiten realizar una composición asistida ya sea usando un repositorio que mediante técnicas de descubrimiento semántico la herramienta pueda realizar recomendaciones.

a) Automáticas

Dentro de las formas de descubrir los componentes existentes podemos encontrar herramientas automáticas y/o asistidas que contribuyen a hacer la labor de la composición mucho más fácil, esto se logra mediante técnicas de recomendación de diferentes tipos, como por ejemplo las semánticas que ofrecen una lista de componentes que pueden ser los apropiados para la aplicación que se está desarrollando.

b) Manuales

Se puede también realizar la tarea de composición de forma manual, es decir eligiendo cada uno de los componentes manualmente de un repositorio o directamente del proveedor del recurso en Internet.

2.1.7.5. Ámbito de los Mashups

Se distinguen desde el grupo objetivo a quienes los Mashups están direccionados. Dirigidas por el mercado de consumo, en los últimos años las herramientas se han centrado más y más en los requisitos empresariales los cuales son distintos a los requisitos del consumidor [26].

a) Mashups Empresariales

Los Mashups Empresariales combinan recursos existentes, sean estos contenidos, datos o funcionalidad de aplicaciones, desde más de una fuente en ambientes empresariales. En contraste a los Mashups de consumo, los ambientes empresariales implican requisitos adicionales como seguridad, calidad o disponibilidad. Adicionalmente, los Mashups empresariales se centran en integrar distintos sistemas. Entonces, los Mashups empresariales tienen enorme potencial para permitir un desarrollo más rápido y a menor costo enfatizando en el desarrollo colectivo, economías de escala, aportando a altos niveles de reutilización y la consecuente habilidad para fabricar opciones apropiadas de software con los datos correctos en el lugar exacto y con el menor tiempo [26].

Los Mashups Empresariales representan una parte en crecimiento de la infraestructura IT en empresas que quiere habilitar a expertos del dominio sin conocimientos de programación. Por ejemplo análisis de datos, para crear aplicaciones para editar o visualizar datos relevantes de negocios [63].

b) Mashups de Consumo

Un Mashup de consumo constituye una herramienta para uso privado como por ejemplo una página de navegador personalizada. Este tipo de Mashups son quizás los más conocidos. Los Mashups de consumo combinan elementos de datos de diferentes fuentes, escondiendo esto detrás de una interface gráfica. En lugar de abrir varias páginas Web para ver por ejemplo el estado del clima, las noticias y los mails privados, el consumidor puede crear una página de inicio uniendo la información desde diferentes fuentes [26].

c) Mashups de Negocios

Algunas veces los Mashups Empresariales y los Mashups de Negocios son diferenciados, si la fuente de contenidos está parcial o totalmente dentro de la empresa, y si la aplicación resultante está destinada a un usuario final o un usuario de negocios [26].

2.1.7.6. Niveles en la Composición de los Mashups

Una de las características más importantes de la Web es su heterogeneidad. Esta heterogeneidad puede ser vista en datos, procesos y hasta interfaces de usuario. Conceptualmente, una aplicación Web Mashup combina información y servicios desde múltiples fuentes de la Web. Generalmente las aplicaciones Web son desarrolladas usando el patrón Modelo-Vista-Controlador (MVC) el cual permite la separación del modelo, presentación y los lógica del negocio. En el patrón MVC, el modelo representa los datos sobre los cuales la aplicación opera y las reglas de negocios usadas para manipular los datos. El modelo es independiente de la vista y el controlador. Este pasivamente provee sus servicios y los datos a las otras capas de la aplicación. La vista representa la salida de la aplicación. Esta especifica como los datos, accedidos a través del modelo son presentados al usuario. Además, mantiene su presentación cuando el modelo cambia. Finalmente el controlador representa la

comunicación entre el modelo y la vista. Este traduce las interacciones en la vista a las acciones ejecutadas sobre el modelo [43].

Se puede concluir entonces que existen tres tipos de salida de un componente que pueden generar una aplicación Web Mashup final: Presentación, datos y visual. El desarrollo de un Mashup puede abarcar uno o todos los niveles de integración incluyendo el nivel de integración de procesos, nivel de integración de datos o integración a nivel de interfaz de usuario, dependiendo de los tipos de salida de la construcción de sus componentes. Además, un solo componente podría tener múltiples salidas [1].

a) Composición a Nivel de Presentación

Cada aplicación necesita una interface para interactuar con los usuarios, y un Mashup no es la excepción. El nivel de presentación (o Interface de usuario) en los Mashups es usada para elicitación de la información de usuario y para mostrar la información final. Las tecnologías usadas para mostrar los resultados al usuario pueden ser simples páginas en HTML o más complejas con Ajax, Java Script, etc. Los lenguajes para integrar componentes de interface y visualizar las salidas soportan Mashups tanto del lado del cliente como del lado del servidor. En los Mashups del lado del Servidor, la integración de los datos y servicios está dada en el servidor, actuando éste como un proxy entre la aplicación del Mashup y otros servicios envueltos en la aplicación. Por otro lado, un Mashup del lado del cliente integra datos y servicios al lado del cliente [43].

b) Composición a Nivel de Datos

Los datos son recuperados desde diferentes fuentes y sistemas. Las herramientas que ofrecen este tipo de composición incluyen conectores desde un amplio rango de fuentes existentes como también desde mecanismos tales como “join” y “filter” [52].

Los componentes que generan datos actúan como fuentes de datos externas, las cuales entregan datos a un Mashup ya sea como cadenas contiguas de datos con propiedades en tiempo real o como snapshots de un conjunto de datos remoto o local. La mayoría de fuentes de datos Web son solo de lectura, pero en algunos casos ellos podrían soportar inclusive actualizaciones [1].

A nivel de datos se accede para integrar datos existentes en múltiples y heterogéneas fuentes tales como datos Web y datos de empresa. Generalmente estos recursos son accesibles a través de REST o servicios Web SOAP, el protocolo HTTP y XML RPC. Recordando que la mediación de datos tiene como problema la estructura y diversidad semántica del esquema a ser usado. Finalmente las fuentes de datos pueden estar estructuradas y bien definidas a través de documentos XML, RSS/ATOM o en forma no estructurada como audio, e-mail, textos, documentos de oficina. Cuando los datos no están estructurados se deben pre-procesar con el fin de extraer su significado y crear datos estructurados. Entonces, este nivel consiste en todas las posibles manipulaciones de datos (conversión, filtro, transformación de formatos, combinación, etc.) necesarios para integrar diferentes fuentes de datos. Cada manipulación podría ser hecha analizando los requisitos sintácticos y semánticos [43].

c) Composición a Nivel de Procesos

Aquí los componentes del Mashup muestran salidas funcionales y son entregados como servicios que contribuyen a la capa de la lógica del negocio del Mashup. Tales componentes son usualmente orquestados [1]. En la integración del Mashup de flujo de procesos los diferentes servicios Web son llamados y orquestados para crear una aplicación compuesta. La aplicación resultante puede ser expuesta como un servicio Web o como una aplicación independiente.

La composición a nivel de proceso ha sido estudiada especialmente en las áreas orientadas al flujo y orientación a servicios. Al nivel de procesos, la coreografía entre las aplicaciones involucradas está definida. La integración es hecha en la capa de aplicación y el proceso de composición es desarrollado combinando actividades, generalmente expuestas a través de APIs. La composición es entonces descrita usando ya sea lenguajes de programación tales como Java o lenguajes de flujo tales como WS-BPEL. En el contexto de los Mashups esos lenguajes no son suficientes para modelar aplicaciones desde, por ejemplo ellos no proveen conexión a los diferentes recursos remotos, como por ejemplo recursos REST y no manejan la interacción con los browsers de los clientes. Estas limitaciones hacen difícil directamente usar estas tecnologías para Mashups, entonces es necesario adaptarlos de acuerdo al modelo y describir los procesos interactivos y asíncronos [43].

2.1.7.7. Maneras de Componer Mashups

Se puede diferenciar entre dos tipos de Mashups según la forma de componerlos: Mashups por agregación y Mashups por integración [16].

a) Agregación

Los Mashups por agregación simplemente ensamblan conjuntos de información desde diferentes fuentes por medio de una interface simple. Los Mashups por agregación no requieren habilidades avanzadas de programación y son a menudo una forma de cortar y pegar de un sitio a otro.

b) Integración

Los Mashups por integración crean aplicaciones más complejas que integran diferentes APIs por ejemplo, para combinar datos desde diferentes fuentes. A diferencia de los Mashups por agregación, el desarrollo de los Mashups por integración necesita considerable experiencia en programación.

2.1.7.8. Herramientas para la Construcción de los Mashups.

Existen una serie de herramientas que permiten al usuario construir Mashups sin necesidad de programarlos o codificarlos. Estas herramientas asisten sustancialmente en la composicionalidad y a menudo traen consigo lenguajes de dominio específicos para Mashups. Como ejemplo de herramientas de desarrollo de Mashups tenemos: Microsoft Popfly, Dapper,

Yahoo Pipes, entre otras. Sin embargo el no tener que escribir código para construir un Mashup no siempre significa que construirlo sea fácil.

Cabe destacar que existen lenguajes de dominio específico (DSL - Domain Specific Languages), que sirven para realizar mashups empresariales, estas herramientas son complejas. Sin embargo existen una serie de estas. Adicionalmente existen otras herramientas que hacen que la construcción de mashups sea algo muy intuitivo.

En la siguiente tabla se presentan algunas herramientas existentes y una descripción breve sobre cada una.

Nombre	Descripción	Características
Yahoo Pipes	Mezcla data feeds para crear Mashups de datos vía un editor visual.	Pipes: Procesa datos de distintas fuentes. Incluye operadores para manipular los data feeds. No proveen interfaces de usuario.
Google Mashup Editor	Obsoleto. Ahora es parte de App Engine, la nueva infraestructura para desarrolladores Web de Google.	
Microsoft Popfly	Componente basado en ambientes visuales para desarrollar Mashups.	
Intel Mash Maker	Ambiente para integrar datos desde páginas Web basadas en un poderoso y dedicado plugin.	Compone Mashups vía copiar y pegar.
QedWiki	Propuesta de IBM para un creador de Mashups basado en wikis.	

Tabla 2-1 Ejemplos de herramientas existentes para la construcción de Mashups.

2.2. Estándares de Calidad Dirigidos a la Usabilidad

Se ha desarrollado una variedad de modelos útiles para especificar y medir la usabilidad del software, la organización internacional para la estandarización ISO ha sido la que ha realizado este tipo de trabajos que han permitido el empleo de estos estándares para realizar métodos de evaluación de usabilidad, lo que constituye uno de los principales objetivos de este trabajo.

Los estándares han sido categorizados en dos grupos: orientados al proceso (ISO 9241 e ISO 13407) y los orientados al producto (ISO 9126 e ISO 14598). Adicionalmente existe una nueva serie de estándares que tratan de mejorar y unificar todos los anteriores (ISO 25000-SQuARE).

Con el rápido desarrollo de las tecnologías de interface de usuario tales como la Web 2.0 y los dispositivos móviles es tentador decir que detrás de un buen diseño existe algo más que la simple aplicación de las normas. Aunque este es el caso, las normas internacionales en la usabilidad tienen un importante rol. Esto se debe a que los estándares de usabilidad:

- **Aseguran la Consistencia:** Los estándares proveen un punto de referencia constante para ayudar a los equipos de diseño y evitar molestas inconsistencias en las interfaces de usuario.
- **Definen Buenas Prácticas:** Existen muchos puntos de vista en conflicto sobre una buena práctica en usabilidad. Los estándares, específicamente los internacionales proveen una guía independiente y con autoridad.

- **Priorizan los Problemas de Interface de Usuario:** Las normas son un asunto serio y que muchas organizaciones prestan poca atención a los resultados de la investigación, pocas empresas pueden permitirse el ignorar las normas.
- **Ayudan a las Organizaciones a Cumplir sus Obligaciones Legales:** leyes de discapacidad y de salud, la legislación de seguridad pone una obligación legal para los proveedores de servicios y los empleadores para garantizar que los sistemas de disposición de los usuarios son aptos para el propósito y cumplir con los requisitos mínimos de ergonomía. Estos requisitos están redactados vagamente en la legislación y cumplir con las normas existentes es una de las mejores formas de demostrar el cumplimiento de las leyes.

2.2.1. Estándares Orientados al Proceso: ISO 9241 e ISO 13407

La norma ISO 9241 está enfocada en la calidad en usabilidad y ergonomía tanto de hardware como software, esta fue creada por la ISO y la IEC en el año de 1999, siendo su principal objetivo regular la calidad de la ergonomía y la usabilidad tanto de hardware como de software. Con el paso del tiempo esta se ha ido mejorando y modificando.

El estándar ISO 92241 es un estándar multi-parte que cubre la parte ergonómica del HCI (interacción hombre sistema). Es uno de los estándares en usabilidad más importantes. Originalmente fue llamado “Conjunto de requisitos para trabajo de oficina y terminales visuales”, este fue concebido como un estándar constituido por 17 partes. Pero despertó interés en los subcomités de estándares en este ámbito, quienes incorporaron otros estándares relevantes e hicieron a este más usable. El título de la revisión del estándar ISO 9241, “Interacción ergonómica hombre – sistema” refleja estos cambios. El estándar revisado está estructurado como una serie de estándares numerados en “cientos” como se presenta continuación [30].

Ámbito	
100	Ergonomía del software
200	Procesos de interacción humano-sistema
300	Pantallas y hardware relacionado a pantallas
400	Dispositivos físicos de entrada – principios ergonómicos
500	Lugar de trabajo ergonómico
600	Medio ambiente ergonómico
700	Dominios de aplicación – Habitaciones de control
900	Interacciones táctiles

Tabla 2-2 Estándares ISO 92941

La ISO 13407 provee una guía para conseguir calidad en uso incorporando actividades diseñadas a través del ciclo de vida de los sistemas interactivos basados en computador. Este describe al diseño centrado en el usuario como una actividad multidisciplinaria, la cual incorpora factores humanos, técnica y conocimiento ergonómico con el objetivo de mejorar efectivamente y productivamente las condiciones de trabajo de los humanos y contrarrestar los posibles efectos adversos del uso sobre la salud de las personas, su seguridad y desempeño [29].

Existen cuatro actividades centradas en el diseño que son necesarias en estados iniciales de un proyecto, y estas son [29]:

- Entender y especificar el contexto de uso.
- Especificar los requisitos del usuario y las organizaciones.
- Producir soluciones de diseño.
- Evaluar diseños de acuerdo a los requisitos.

Estas actividades son iterativas y el nivel de esfuerzo en detalle varía dependiendo el ambiente de diseño y el estado del proceso de diseño [29].

Existen ventajas de emplear este conjunto de normas son:

- Se identifican aspectos de usabilidad y componentes del contexto de uso que deben tenerse en cuenta en las fases de especificación de requisitos, diseño y evaluación de usabilidad.
- El rendimiento y la satisfacción del usuario proporcionan medidas directas de usabilidad en un contexto dado.
- Las medidas que se obtienen dan una base para comparar la usabilidad de varios diseños en un mismo contexto.

Sin embargo algunos de los puntos débiles en este conjunto de normas serían [19]:

- La usabilidad está estrictamente tratada desde una perspectiva de proceso, no centrada en el producto.
- No se tiene en cuenta la “facilidad de aprendizaje”, siendo esta recomendada por otros estándares y expertos en usabilidad.
- No tiene en cuenta aspectos de seguridad.

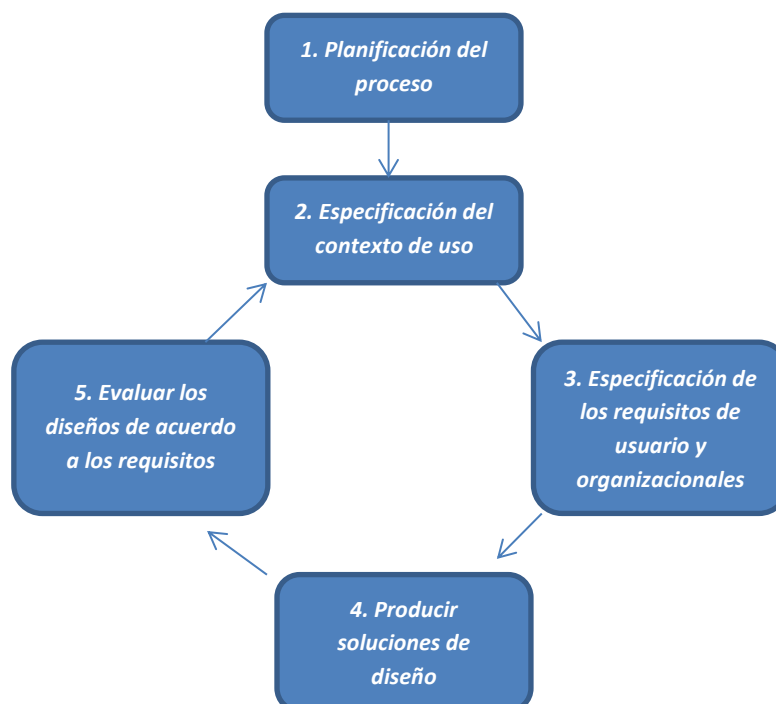


Figura 2-1

Actividades en el diseño del método de evaluación

2.2.2. Estándares Orientados al Producto: ISO 9126 e ISO 14598

El estándar ISO 9126 [32], es un estándar internacional de evaluación de calidad de software. Este estándar ha sido reemplazado por el proyecto SQuaRE, el cual sigue los mismos conceptos. El estándar está dividido en cuatro partes:

ISO/IEC 9126-1: Esta parte presenta el Modelo de Calidad en base a dos vistas que marcan el ciclo de vida de la calidad:

- Calidad interna y externa, las cuales se modelan con el mismo conjunto de características: funcionalidad, fiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad.

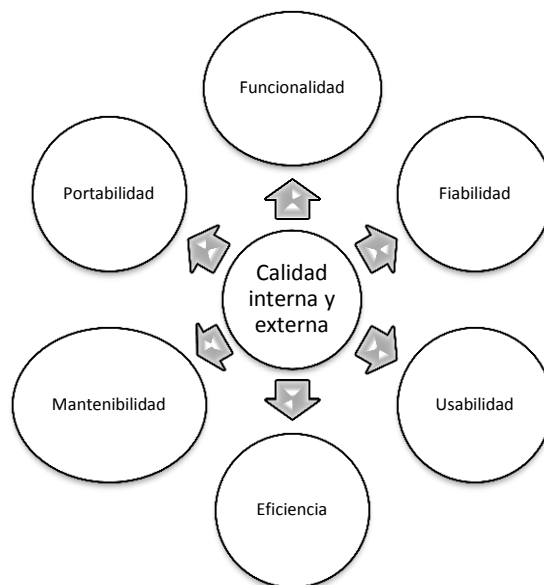


Figura 2-2 *Calidad interna y externa.*

- Calidad en uso modelada con otras cuatro características: eficacia, productividad, seguridad y satisfacción.

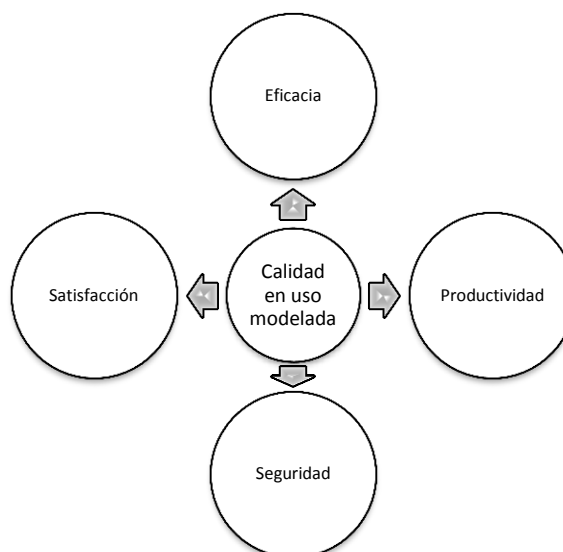


Figura 2-3 *Calidad en uso Modelada*

- ISO/IEC 9126-2: Esta parte describe las métricas externas que son utilizadas para especificar o evaluar el comportamiento del software cuando es operado por el usuario.
- ISO/IEC 9126-3: Esta parte describe las métricas internas que se pueden utilizar para crear describir propiedades estáticas de la interfaz, que puede ser evaluadas por la inspección sin poner en funcionamiento el software.
- ISO/IEC 9126-4: Esta parte describe las métricas de calidad en uso que se pueden utilizar para especificar o evaluar el efecto del producto software cuando son operados por el usuario en determinados contextos de uso.

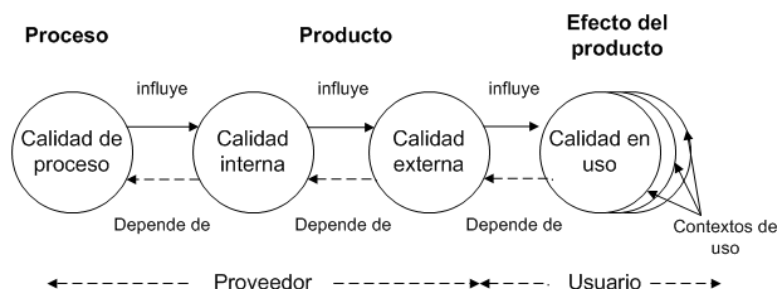


Figura 2-4 *Calidad Proceso, Producto y En Uso*

El concepto de usabilidad fue definido según el ISO 9126 en su versión del año 1991 como “el conjunto de atributos que influyen en el esfuerzo necesario para su utilización y en la evaluación individual de tal uso, por un conjunto de usuarios”. Como se puede ver en la definición el concepto de usabilidad, está directamente relacionado al producto. Posteriormente en la versión del 2001, se hace un alcance más preciso a lo que la usabilidad abarca, mencionando que la usabilidad es “La capacidad que tiene un producto software para ser entendido, aprendido, operable, atractivo para el usuario y conforme a estándares/guías, cuando es utilizado bajo unas condiciones específicas”.

La ISO 9126, se limita a la especificación de un modelo de calidad general, mientras que la ISO 14598 proporciona un marco de trabajo que evalúa la calidad de todo tipo de software e indica los requisitos para los métodos de medición y el proceso de evaluación.

ISO/IEC 14598-1	Visión general de todo el estándar y explicación de las diferencias entre la evaluación del producto software y el modelo de calidad definido en la ISO / IEC 9126.
ISO/IEC 14598-2	Requisitos y guías para las funciones de planificación y gestión de la evaluación del producto.
ISO/IEC 14598-3	Requisitos y guías para la evaluación del producto software cuando la evaluación se lleva a cabo en paralelo al desarrollo del mismo.
ISO/IEC 14598-4	Requisitos y guías para la evaluación del producto software cuando este ha sido adquirido y se requiere reutilizar un producto existente o pre-desarrollado.
ISO/IEC 14598-5	Requisitos y guías para la evaluación del producto cuando esta es llevada a cabo por evaluadores independientes.
ISO/IEC 14598-6	Provee las guías para la documentación del módulo de evaluación.

Tabla 2-3 *Partes del estándar ISO / IEC 14598*

Este trabajo trata no solamente de la elaboración de un modelo de calidad sino también del proceso de evaluación del producto software, por tanto se ha tenido en cuenta no solo

estándares que sirvieron de base para elaborar el modelo sino también aquellos que permitan seguir el proceso de la evaluación. Entonces, el estándar ISO/IEC 14598, fue la primera aproximación que se utilizó para llevar a cabo este proceso, este consta de seis partes según consta en la Tabla 2-3.

2.2.3. Estándar ISO 25000 (SQuaRE)

La meta perseguida en la creación de esta norma es dar un paso hacia un conjunto de estándares organizados de manera más lógica, enriquecidos con nuevas aportaciones y unificados con respecto a las normas anteriores para ser capaces de cubrir los dos principales procesos: especificación de requisitos de calidad del software y evaluación de la calidad del software, soportados por un proceso de medición. SQuaRE se centra exclusivamente en el producto software estableciendo criterios para su especificación, su medición y su evaluación. Es decir, básicamente se trata de una unificación y revisión de los estándares ISO/IEC 9126 e ISO/IEC 14598.

La comprensión de la especificación y la evaluación de la calidad de software y sistemas de cómputo, es un factor que asegura valor a los stakeholders. Esto puede ser conseguido definiendo las características de calidad deseadas y necesarias asociadas con los stakeholders, sus metas y objetivos para el sistema. Esto incluye las características de calidad relacionadas al sistema de software y los datos, como también, el impacto que el sistema tiene en los stakeholders.

El SQuaRE define seis características de calidad y describe un modelo de proceso de evaluación de producto software. SQuaRE por tanto revisa el ISO/IEC 9126-1:2001, e incorpora las mismas características de calidad de software con algunas enmiendas.

- El ámbito de los modelos de calidad ha sido extendido para incluir sistemas de computación y calidad en uso desde la perspectiva del sistema.
- El contexto cubierto ha sido añadido como una característica de calidad en uso, con subcaracterísticas de completitud de contexto y flexibilidad.
- La seguridad ha sido añadida como una característica, además de una subcaracterística de funcionalidad, con subcaracterísticas de confidencialidad, integridad, no repudio, contabilidad y autenticidad.
- Compatibilidad (incluyendo interoperabilidad y co-existencia) ha sido añadida como una característica.
- Las siguientes subcaracterísticas han sido añadidas: completitud funcional, capacidad, protección de errores de usuario, accesibilidad, disponibilidad, modularidad y reusabilidad.
- Las subcaracterísticas de conformidad han sido removidas, como conformidad con leyes y regulaciones.
- Los modelos de calidad interna y externa han sido combinados como el modelo de calidad de producto.
- Cuando es apropiado, las definiciones genéricas han sido adoptadas, además usando las definiciones específicas de software.
- Se han dado nombres más precisos a algunas características y sub-características.

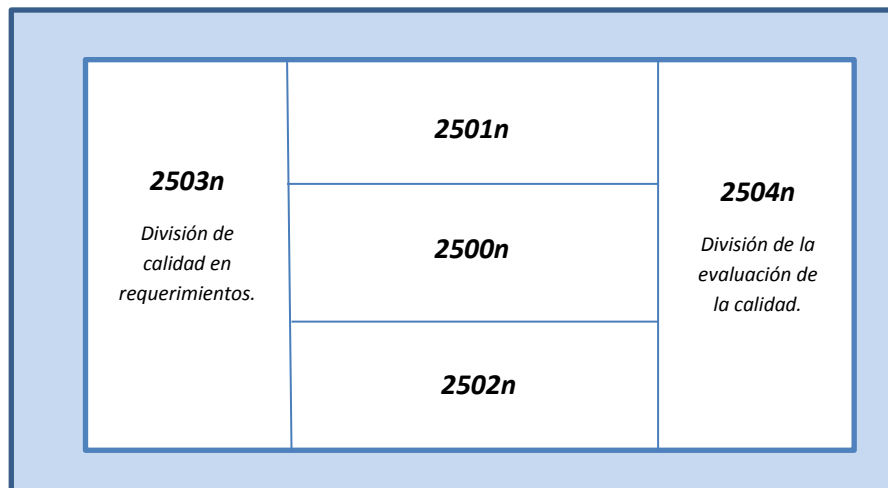


Figura 2-5 Organización de las series SQuaRE

Las divisiones del SQuaRE son:

- **ISO/IEC 2500n** – División de Manejo de la calidad. Los Estándares internacionales que forman esta división definen todos los modelos comunes, términos y definiciones además referidos a todos los otros estándares internacionales del SQuaRE. La división además provee requisitos y guías para una función de soporte que es responsable del manejo de los requisitos, especificación y evaluación de la calidad de producto del software.
- **ISO / IEC 2501n** – División del modelo de calidad. Los Estándares Internacionales que forman esta división presentan modelos de calidad detallados para sistemas de cómputo y productos de software, calidad en uso y datos. Una guía práctica del uso de los modelos de calidad es además provista.
- **ISO / IEC 2502n** – División de Medida de la Calidad. Los estándares internacionales que forman esta división incluyen un modelo de referencia para la medir la calidad del producto, definiciones matemáticas o medidas de calidad y guías prácticas para su aplicación. Presenta métricas aplicadas a la calidad interna del software, calidad externa de software y calidad en uso.
- **ISO / IEC 2503n** – División de calidad de requisitos. El estándar que forma esta división ayuda a especificar los requisitos de calidad. Estos requisitos de calidad pueden ser usados en el proceso de elicitación de requisitos de calidad, para un producto de software que puede ser desarrollado o como entrada de un proceso de evaluación. El proceso de definición de requisitos es mapeado a procesos técnicos definidos en ISO / IEC 15288 – Tecnología de la información – Manejo del ciclo de vida – Proceso del sistema de ciclo de vida.
- **ISO / IEC 2504n** – División de la calidad de la evaluación. Los estándares que forman esta división proveen requisitos, recomendaciones y guías para la evaluación del producto de software ya sea ejecutada por evaluadores independientes, adquirientes o desarrolladores. Además, presenta el soporte para documentar una medida como un módulo de evaluación.
- **ISO/IEC 25050-25099** – División de extensión del SQuaRE. Estos Estándares internacionales, actualmente incluyen requisitos para la calidad de software que servirán para reportes de usabilidad.

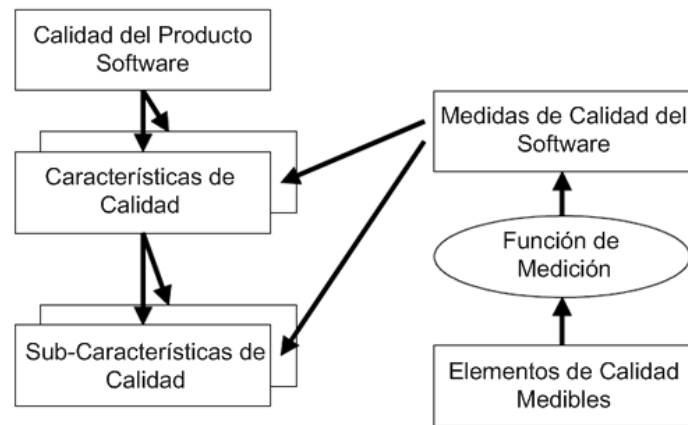


Figura 2-6 *Modelo de medición de la calidad del producto software según SQuaRE.*

2.3. Modelo de Referencia para la Evaluación de los Productos de Software

El modelo de referencia de evaluación de la calidad de producto software describe las entradas y salidas, restricciones y recursos para el proceso de evaluación del producto de software, en el caso de los Mashups el modelo de referencia general ha sido la base para elaborar el proceso de evaluación específico para este tipo de aplicaciones.

El modelo de referencia para la evaluación de la calidad de un producto de software se aplica a los responsables de la evaluación del mismo. Es apropiado para organizaciones en su rol de adquirientes, desarrolladores o evaluadores. Se dirige pero no se limita a los mismos.

Para el caso de los Mashups, los roles pueden ser los mismos, sin embargo la peculiaridad de este tipo de aplicaciones en la mayor de las veces se limita al desarrollador y al usuario final, que incluso puede llegar a ser la misma persona.

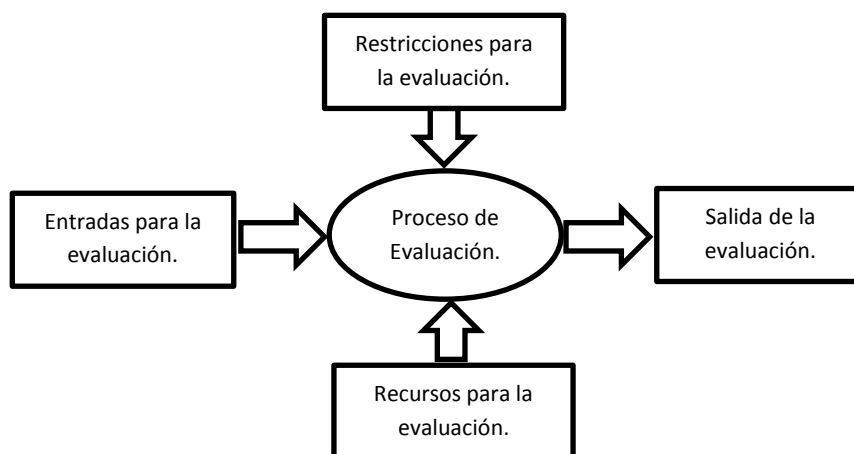


Figura 2-7 *Proceso de evaluación para productos de software*

2.3.1. Entradas de la Evaluación

Las entradas para la evaluación de la calidad del Mashup incluyen lo siguiente:

- Requisitos para la evaluación de la calidad del Mashup.

- Especificación de los requisitos de calidad del producto software (ver ISO/IEC 25030).
- Producto software y productos intermedios a ser evaluados.

Para el caso de los Mashups, tendremos en cuenta qué requisitos necesita cubrir el usuario, al ser la naturaleza del Mashup “situacional” y muchas veces el usuario final tendrá el rol de desarrollador, este tendrá muy en claro el dominio del problema, al momento de la especificación de los requisitos de calidad también tendrá una idea mucho más cabal de qué es lo que necesita, cuáles serán los requisitos de calidad que espera del producto (especificación de los requisitos de calidad del producto de software), cuáles serán los factores que influirán en la selección de los componentes (productos intermedios a evaluarse) y además qué requisitos de evaluación de la calidad se tendrán en cuenta.



Figura 2-8

Entradas de la evaluación de los Mashups

2.3.2. Salidas de la Evaluación

Como un resultado de la implementación exitosa del proceso de evaluación de la calidad del producto software tenemos:

- Un plan de evaluación de la calidad del producto software.
- Los requisitos de la calidad del producto software son especificados y priorizados.
- Las métricas de calidad son seleccionadas.
- El criterio de decisión definido para las métricas.
- El criterio de decisión para la evaluación.
- Las actividades de la evaluación son planificadas.
- Las métricas de calidad son obtenidas.
- La calidad del producto software evaluada.
- Los resultados de la evaluación son revisados.
- Un reporte de la evaluación del producto de software es producida.
- Los datos de la evaluación.

- l) Una vez ejecutada la evaluación obtendremos las salidas anteriormente citadas, en el caso de los Mashups tendremos un plan de evaluación de la calidad del Mashup, adicionalmente los requisitos de la calidad del Mashup serán especificados y priorizados, teniendo en cuenta su carácter composicional, así como también sus características específicas, las métricas de calidad serán seleccionadas conjuntamente con la definición de las métricas. El criterio de decisión para la evaluación y las actividades de la evaluación serán planificadas, esta evaluación aportará tanto en el momento de componer el Mashup como también en el producto final ayudará a tomar las medidas correctivas para conseguir un producto de alta calidad. Las actividades de la evaluación serán planificadas y las métricas de calidad serán obtenidas, evaluándose así de una manera poco subjetiva la calidad del Mashup y obteniéndose el reporte de la evaluación conjuntamente con los datos obtenidos en el proceso.



Figura 2-9 *Salidas para la evaluación*

2.3.3. Restricciones para la Evaluación.

Las restricciones para la evaluación de la calidad del producto software incluyen lo siguiente.

- Restricciones de las necesidades de la evaluación de la calidad del producto software.
- Restricciones de los recursos de la evaluación de la calidad del producto software.
- Restricciones de los horarios de la evaluación de la calidad del producto software.
- Restricciones de coste de la evaluación de la calidad del producto software.
- Restricciones de ambiente de la evaluación de la calidad del producto software.
- Restricciones de herramientas y metodología de la evaluación de la calidad del producto software.

g. Restricciones de reportes de la evaluación de la calidad del producto software.

Las restricciones para la evaluación de los Mashups se deben fijar considerando los puntos anteriormente citados y plasmados en el gráfico. Por el carácter composicional de los Mashups, se debe tener en cuenta que para la evaluación se debe contar con que los componentes estén operativos al momento de la misma y de no ser así este podría ser un punto influyente en la usabilidad del producto y en la calidad en general. En este trabajo se aborda exclusivamente la usabilidad de los mashups, por tanto se tendrán en cuenta todas las restricciones que impliquen la usabilidad de estas aplicaciones. Para ello se ha especificado un modelo de calidad de usabilidad en capítulos anteriores que es usado para delimitar el proceso de evaluación, en cuanto a las características y métricas a ser evaluadas.

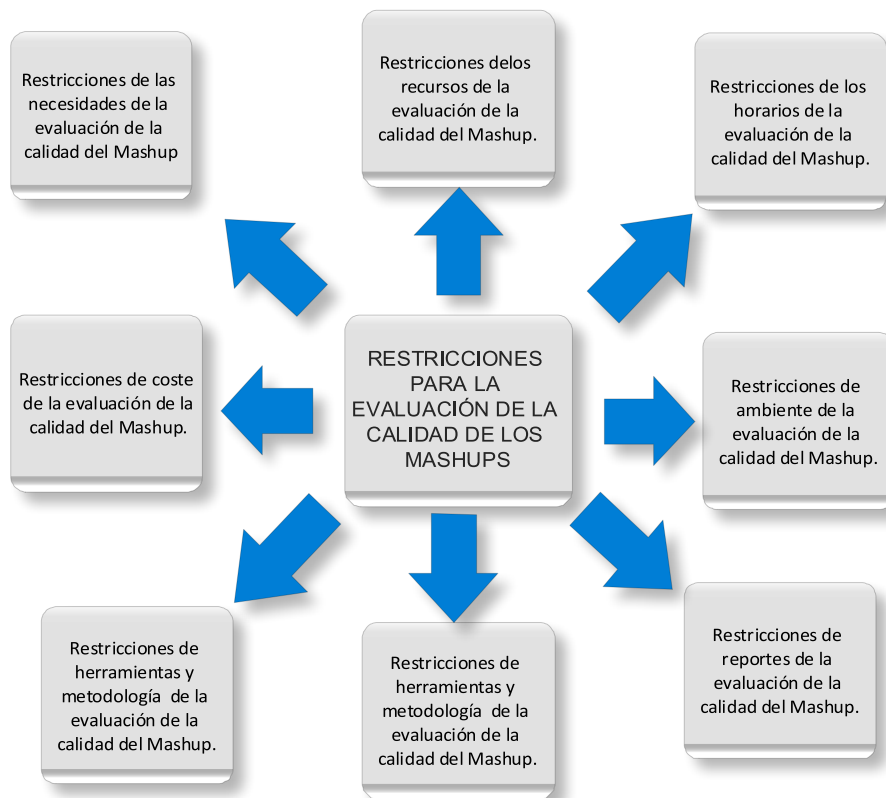


Figura 2-10 Restricciones para la evaluación

2.3.4. Recursos para la evaluación.

Los recursos para el proceso de evaluación de la calidad del producto software incluyen lo siguiente:

- Herramientas de medidas aplicables y metodología, incluyendo módulos de evaluación.
- Documentos aplicables del SQuRE (ISO/IEC 25001, 25010, 2502n, 25041).
- Recursos humanos para la evaluación de la calidad del producto software.
- Recursos económicos para la evaluación de la calidad del producto software.
- Sistema de información para la evaluación de la calidad del producto de software.

- Base de datos de conocimiento para la evaluación de la calidad del producto de software.

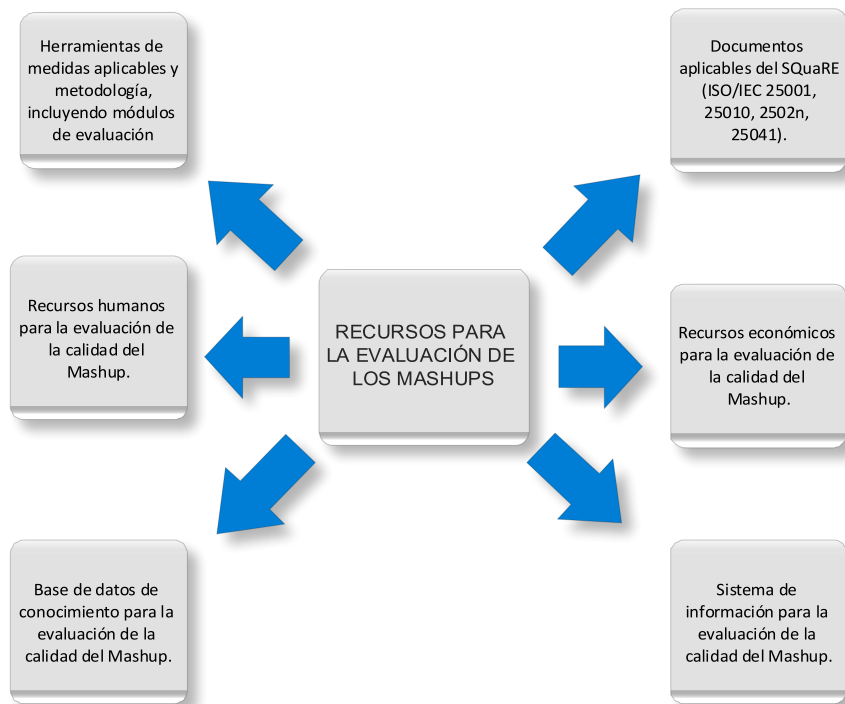


Figura 2-11 Recursos para la evaluación

En la evaluación se ven involucrados un conjunto de recursos necesarios para llevarla a cabo tales como las herramientas de medidas aplicables, la metodología y los módulos de evaluación, además se deben considerar todos los documentos que se puedan aplicar, tenemos estándares que permiten definir el modelo de calidad del producto, las métricas a utilizar para la evaluación, etc. Los recursos humanos que ejecutarán la evaluación de la calidad del Mashup, en este caso podrá ser el desarrollador que en muchos casos también actúa como usuario final, facilitando la tarea de la evaluación por el grado de entendimiento del dominio del problema ya que cualquier cambio o proceso de mejora será más entendido, también vale la pena recordar que el Mashup generalmente trata de una sola página por lo que mayoritariamente no necesitamos el concepto de navegabilidad, acotando en muchos casos la cantidad de recursos humanos necesarios para la evaluación. En cuanto los recursos económicos para la calidad del Mashup, también es importante considerar las necesidades específicas de este tipo de aplicaciones, la conectividad necesaria, ancho de banda, etc. La base de datos de conocimiento para la evaluación de la calidad del Mashup se construirá a lo largo de la evaluación y adicionalmente se puede tener en cuenta ciertas valoraciones con respecto a cuestiones de los componentes tales como accesibilidad, ranking de disponibilidad, popularidad de componentes, etc, que permitirán ir creando una base de datos de conocimiento para futuras evaluaciones. Por último un sistema de información para la evaluación de la calidad del mashup, se tendrá en cuenta como un trabajo futuro.

Roles

Distintos roles incluyendo adquirientes, desarrolladores, evaluadores independientes, proveedores, operadores y las personas que realizan el mantenimiento tienen diferentes propósitos en el desarrollo de un producto de software, sin embargo para el caso de los mashups nos limitaremos a tomar solamente el punto de vista del adquiriente, usuario (operador) y del desarrollador, al ser aplicaciones que generalmente abarcan una única página, generalmente no se hace necesaria la intervención de los evaluadores independiente, personas que realizan el mantenimiento y otros aunque si los hubiere, se podrían utilizar los principios de evaluación de productos de software en general que se encuentran a lo largo del SQuaRE:

- **El Adquiriente:** Cuando se adquiere un producto de software en este caso un Mashup hecho a la medida, el adquiriente establece los requisitos de calidad en uso y los requisitos de la calidad del producto software, especifica los requisitos al proveedor y evalúa las compras potenciales contra esos requisitos antes de la adquisición. Por la condición de aplicaciones situacionales, de los mashups, nos encontraremos no muy frecuentemente con este rol, cuando se adquiere un Mashup a ser desarrollado, el objetivo de especificar los requisitos de calidad es asegurar que el producto reúne las necesidades de calidad del usuario, la evaluación puede ser usada para comparar productos alternativos y para asegurar que el Mashup seleccionado reúne los requisitos de calidad.
- **El Desarrollador:** En el caso específico de los Mashups, se debe tener en cuenta que muchas veces el rol del desarrollador se ve directamente involucrado al del usuario final ya que suele tratarse de la misma persona, el desarrollador debería evaluar los productos de software intermedios o el producto final para asegurar la calidad del software desarrollado. El desarrollador puede usar los resultados de la evaluación del producto para asegurar que productos reúnen los criterios de calidad, los cuales pueden ser fijados por el adquiriente, o por la comparación con otros productos.
- **El Operador:** Es el individuo u organización, quien opera con un Mashup, puede usar la evaluación de la calidad el software para validar que los requisitos de calidad son reunidos bajo condiciones de operación variables y proveen retroalimentación sobre la necesidad de cualquier cambio bajo su responsabilidad que pueda darse en razón de mantenimiento.

Calidad en el ciclo de vida.

En la Figura 2-12 Calidad en el ciclo de vida de un producto de software, se muestran diferentes enfoques para especificar requisitos y evaluar la calidad del Mashup.

- La calidad de uso hace referencia a la calidad del Mashup en su contexto de uso.
- Un Mashup como una parte de un sistema en operación se refiere a las medidas externas de calidad de software. En este caso generalmente son aplicaciones aisladas que si bien forman parte muchas veces del conjunto de programas que posee una empresa por ejemplo, generalmente no están vinculadas al sistema transaccional de la empresa.

- Los artefactos estáticos de un Mashup se relacionan a las medidas internas de calidad de software. En el caso de los Mashups, podríamos considerar que si se hace una medición de calidad sobre un componente aislado, se lo vería como una medida interna de calidad.

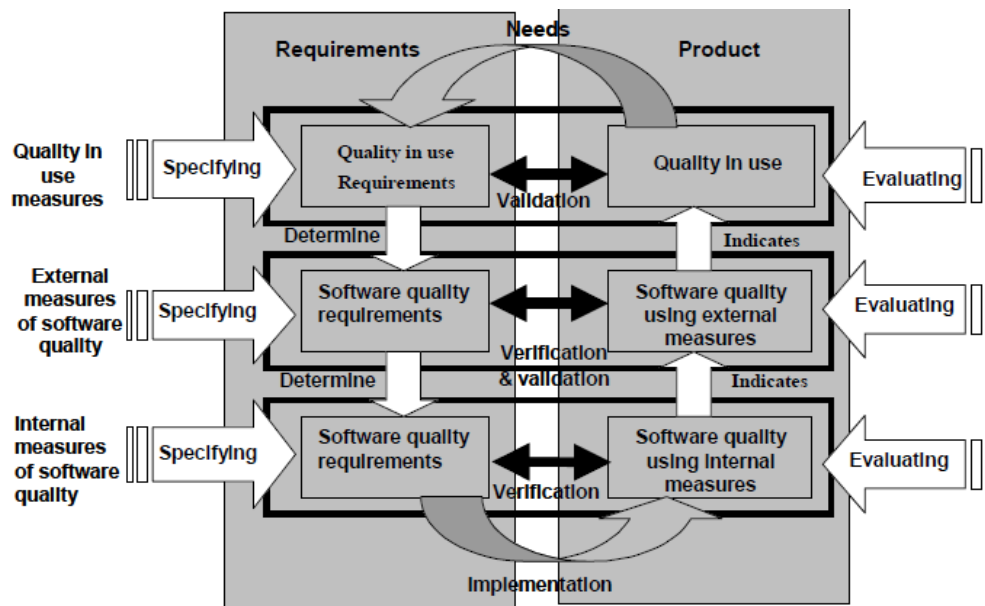


Figura 2-12 *Calidad en el ciclo de vida de un producto de software*

El objetivo es que cuando el producto de software es actualmente usado por el usuario este reúne las necesidades fijas e implícitas. Las medidas externas de calidad de software pueden solamente ser aseguradas para un sistema/subsistema en el cual el Mashup es una parte. Las medidas externas son aplicadas cuando se ejecuta el software. Los valores de las medidas externas necesariamente dependen sobre más que el software, entonces el software tiene que ser evaluados como parte de un sistema trabajado, siendo esto muy poco usual en el tipo de aplicaciones que en este trabajo se abordan.

La calidad en uso es el efecto combinado sobre las características de calidad relevantes para un usuario particular. El software, el cual se ejecuta satisfactoriamente en un ambiente, puede fallar en otro ambiente. La evaluación externa de las características de calidad de ahí toma lugar bajo condiciones que emulan tan cerca como sea posible las condiciones esperadas para el uso. Las medidas externas de características son hechas cuando el código está completo, tanto como si no es posible emular las condiciones exactas de uso (ejm. Redes y características de usuario), las medidas externas son a menudo solamente indicadores de la calidad en uso actual.

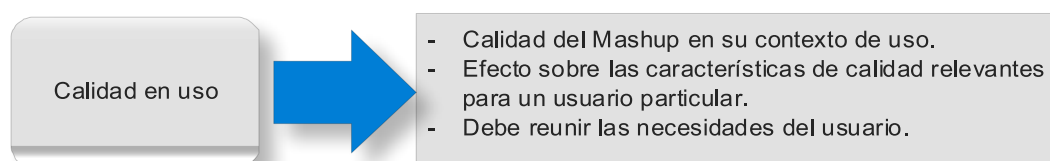


Figura 2-13 *Características calidad en uso*

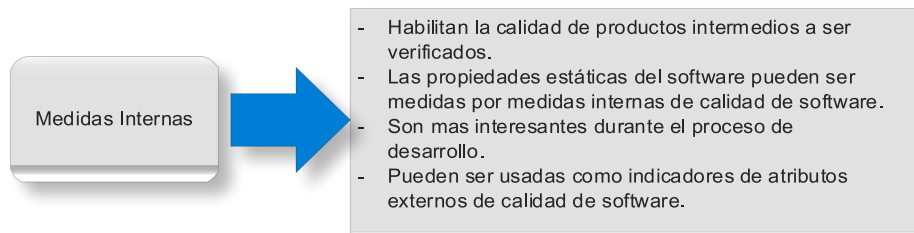


Figura 2-14 *Características de las medidas internas de calidad*

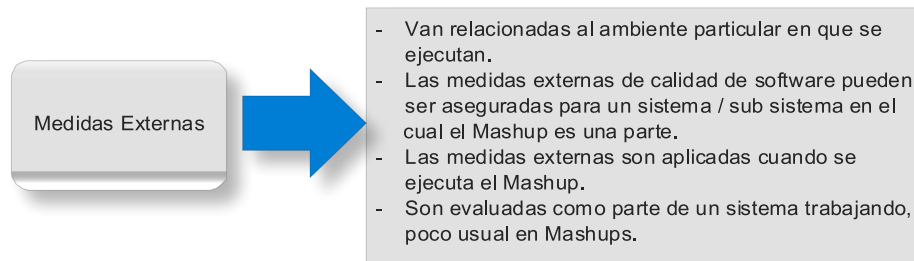


Figura 2-15 *Características de las medidas externas de calidad*

Los resultados de una evaluación pueden ser usados como retro alimentadores para modificar las características del software y así mejorar la calidad de software, en un proceso iterativo de mejora.

Para los propósitos de desarrollo, los requisitos para las medidas internas de calidad de software son definidos, los cuales habilitan la calidad de productos intermedios a ser verificados por ejemplo para el caso de un Mashup, un producto intermedio puede ser un componente. Las propiedades estáticas (ej. La especificación o código fuente) del software pueden ser medidos por medidas internas de calidad de software. Las medidas internas de calidad de software son más interesantes durante el proceso de desarrollo. Conseguir las medidas de calidad interna requeridas puede contribuir a reunir los requisitos para medidas externas de calidad de software.

Es muy importante que las medidas de atributos de calidad de software sean relacionadas a requisitos para medidas externas de calidad de software, entonces las características de calidad de productos de software bajo desarrollo (ambas intermedias y finales) pueden ser aseguradas con respecto a las necesidades de calidad final del sistema.

En el caso de los Mashups las medidas externas de calidad de software podrían ir relacionadas al ambiente particular en el que se ejecutan; por ejemplo, por el ancho de banda, por su disponibilidad, etc.; es decir si se mide considerando todo el Mashup en global, aspectos tales como la carga, disponibilidad, etc.; podrían ser consideradas medidas de calidad externas.

Soporte para la Evaluación.

Las siguientes son actividades para asistir la evaluación para recolectar información sobre la calidad de un Mashup, métodos y herramientas, desarrollando y validando métricas y estandarizando el proceso de evaluación y medición.

ISO / IEC 25001 contiene requisitos y guías para soportar los procesos para evaluar la calidad de producto de software como también para mejorar el nivel organizacional lo que significa por ejemplo evaluar cada proyecto de evaluación y el propio proceso de evaluación.

ISO / IEC 25041 define la estructura y contenido de los módulos de evaluación. Los documentos de evaluación tanto los de evaluación de la tecnología como lo de los procedimientos para aplicar la tecnología.

Capítulo 3. Estado Actual de la Investigación

En este capítulo se muestran los estudios que se han realizado a lo largo de esta tesina para de esta manera tener una visión global sobre la evaluación de la usabilidad en el ámbito de los Mashups y su composicionalidad.

En primer lugar se detallará el método aplicado para la recolección de la información que será útil en el presente capítulo.

En la sección 3.2 se presenta un mapeo sistemático sobre la calidad del producto en los Mashups, en donde se han analizado aspectos relacionados a la calidad del producto, en qué estado del desarrollo esta se ve envuelta la calidad del producto, cuáles son los artefactos involucrados en los estudios de la calidad de los Mashups, el tipo de enfoque, el tipo de validación de los estudios y por último en qué ámbito se usan estas aplicaciones.

En la sección 3.3 se presenta un mapeo sistemático sobre la composicionalidad de los Mashups, en donde se analizan aspectos relacionados al carácter particular de este tipo de aplicaciones Web.

3.1. Introducción a los Estudios de Mapeos Sistemáticos

Un mapeo sistemático es una metodología frecuentemente usada en investigación médica y que ha sido recomendada para áreas de investigación donde existe una falta de estudios primarios relevantes de alta calidad [36].

Un mapeo sistemático en el área de la ingeniería del software es un método para construir un esquema de clasificación y estructurar un campo de interés en la ingeniería del software. El análisis de los resultados se centra en frecuencias de publicaciones por categorías [53].

Los resultados de un mapeo sistemático pueden identificar áreas en las cuales es necesario conducir una revisión sistemática.

Las principales diferencias entre un mapeo sistemático y una revisión son [36]:

- Los mapeos son generalmente manejados por preguntas de investigación extensas y a menudo responden múltiples preguntas de investigación.
- Los términos de búsqueda para mapeos sistemáticos suelen ser enfocados a un nivel más alto que las revisiones sistemáticas y retornan un gran número de estudios. Para un mapeo, sin embargo, es menos problemático trabajar con abundantes resultados.
- El proceso de extracción de datos para los mapeos sistemáticos es además mucho más amplia que la extracción de datos para revisiones sistemáticas y puede ser definido como un estado de clasificación o categorización. El propósito de este estado es clasificar los estudios con suficiente detalle para responder las preguntas de investigación más amplias y para identificar estudios para las subsecuentes revisiones.
- El estado de análisis para un estudio de mapeo sistemático consiste en resumir los datos de acuerdo a responder la pregunta de investigación. No es necesario incluir técnicas de análisis a profundidad tales como el meta-análisis y la síntesis narrativa,

sino totales y resúmenes. Representaciones gráficas de distribución de estudios por tipos de clasificación pueden ser un tipo efectivo de mecanismo de presentación de resultados.

- La diseminación de los resultados de un mapeo puede ser más limitada que de una revisión sistemática.

Los pasos esenciales de un mapeo sistemático son la definición de la pregunta de investigación, la conducción de la búsqueda de estudios relevantes, los artículos, las palabras claves, los resúmenes y la extracción de datos y mapeos. Cada proceso tiene una salida, el resultado obtenido será un mapeo sistemático.

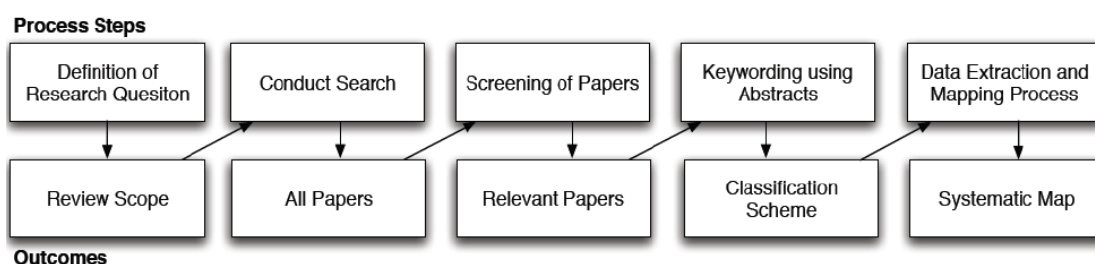


Figura 3-1 Pasos de un Mapeo Sistemático

Una vez revisado el estado del arte, se puede pasar a realizar la propuesta en base a los resultados obtenidos y se puede dar una idea clara del porqué del trabajo propuesto y de los gaps que existentes que podrían conllevar a investigaciones futuras.

3.2. Mapeo Sistemático sobre la Calidad de Producto en Mashups

Con el fin de elaborar un estado del arte preciso se ha realizado una exhaustiva investigación acerca de los aspectos de calidad más relevantes en el ámbito de los Mashups. Para ello se ha empleado un método que permita ubicar toda la información actual mediante un Mapeo Sistemático.

Método de Investigación

Se ha elaborado un mapeo sistemático considerando las recomendaciones provistas en trabajos de [36] y [53].

Nuestro mapeo sistemático ha sido ejecutado en tres estados: planificación, conducción y reporte de resultados.

3.2.1. Fase de Planificación

El reto de desarrollar Web Mashups ha motivado la aparición de una variedad de técnicas, métodos y herramientas para direccionar su proceso de composición. Sin embargo, algunos aspectos aún no han sido explorados, como son los aspectos de calidad específica de este tipo de aplicaciones Web [14], siendo necesario el estudio de estas características específicamente para Mashups.

Hasta donde se conoce, no existen estudios que indiquen como los aspectos de calidad en los Mashups han sido direccionados. En esta sección, nosotros presentamos un mapeo

sistemático que categoriza y resume la información existente sobre una pregunta de investigación en una manera imparcial. La meta de este apartado es de conocer cómo los aspectos de calidad han sido dirigidos en el desarrollo de los Mashups y cómo estos han sido validados.

Un número de investigaciones y revisiones tratan de analizar los enfoques de desarrollo de Mashups actuales, enfoques y herramientas han sido reportadas en años recientes [9], [22], [26][52].

Hoyer y Fisher [26], presentaron una revisión de las diferentes herramientas para Mashups existentes en el mercado, las han clasificado y evaluado de acuerdo a algunas perspectivas tales como información general, funcionalidad y usabilidad. Ellos clasificaron más de 30 herramientas en su modelo de clasificación propuesto, para de este modo ver las tendencias del mercado en el contexto de los Mashups Empresariales. Sin embargo, la evaluación de los aspectos de calidad estaba relacionada al proceso de desarrollo más que a la calidad del producto de los Mashups generados.

Beemer y Green [9], presentaron una revisión basada en 60 publicaciones que ayuda a los investigadores a clasificar los tópicos generales de investigación sobre Mashups. Una clasificación genérica fue definida de acuerdo a seis categorías: control de acceso, integración, agentes, marcos de trabajo, programación de usuario final y Mashups Empresariales. Además una revisión metodológica fue establecida y conducida, aspectos de calidad de Mashups no se identificaron como un tópico de investigación específico.

Grammel y Storey [22], presentan una visión en conjunto del desarrollo de Usuario Final (End-User development) en un conjunto seleccionado de ambientes de desarrollo. Ellos exploraron, resumieron y compararon sus características a través de seis diferentes aspectos: niveles de abstracción, soporte de aprendizaje, soporte de comunidad, detección, diseño de interfaces de usuario y técnicas de ingeniería de software. Los resultados mostraron que existe aún mucho para futuras mejoras. Nuevamente los autores no discutieron los aspectos de calidad de producto de los Mashups.

El laboratorio Orange Labs [52], presentó una investigación que trata de evaluar herramientas existentes para Mashups como también para refinar los requisitos de diseño para desarrollar esas herramientas. Este trabajo muestra explícitamente la necesidad de enfoques de evaluación de calidad, desde que esto fue publicado en 2009 [3], el 62% de CIOs no fueron vistos confiables como las aplicaciones Web tradicionales y no se adaptaron a empresas.

A pesar de algunos estudios y revisiones reportadas, estas presentaron dos limitaciones principales.

Aquí es necesario un camino más sistemático para resumir el conocimiento existente en esta área, ya que la mayoría de estos estudios son informales debido a que no definen preguntas de investigación, procesos de búsqueda, procesos de extracción de datos o de análisis de datos.

Se hacen necesarios estudios o revisiones específicamente centradas en la calidad del producto de los Mashups. Nosotros estamos conscientes de que existen algunos estudios que abordan los problemas de calidad en los Mashups tales como [12], [54], [58]. Sin embargo,

como sabemos, no existen estudios empíricos basados en evidencia que reporten la categorización y resumen de esos estudios.

Pregunta de Investigación

Nosotros hemos llevado a cabo el mapeo sistemático, considerando las guías sugeridas en [36] y [53]. La meta de nuestro estudio es examinar cuáles y cómo los aspectos de calidad de los Mashups han sido dirigidos desde el punto de vista de las siguientes preguntas de investigación:

RQ1: ¿Cuál / cómo los problemas de calidad han sido abordados en el desarrollo de los Mashups?

RQ2: ¿Cómo los enfoques de calidad de producto han sido definidos y validados?

Estas preguntas de investigación nos permitirán resumir el conocimiento actual sobre la calidad de producto de los Mashups e identificar lagunas en la investigación actual para así sugerir áreas para investigaciones futuras.

Identificación y Selección de Estudios Primarios

Las principales fuentes que hemos usado para la búsqueda de estudios primarios son las librerías digitales IEEE Xplore y ACM.

Adicionalmente, nosotros hemos buscado manualmente en las ediciones de las siguientes revistas, libros y conferencias:

- Journal of Web Engineering (JWE).
- ACM Transactions on the Web (ACMTWEB).
- Foundations of Popfly: Rapid Mashup Development (Book).
- World Wide Web conference (WWW).
- International Conference on Web Engineering (ICWE).
- International Conference on Information Integration and Web-Based Applications & Services (iiWAS).
- International Conference on Service Oriented Computing (ICSOC).
- International Workshop on Web APIs and Service Mashups (MASHUPS).

La cadena de búsqueda definida para obtener los estudios es la siguiente: “(Web OR internet OR www) AND (mash*) AND quality”. El símbolo (“*”) permitirá cualquier variación de palabra en cualquier cada término de búsqueda que lo contenga (ej. La búsqueda del término “mash*” incluye las siguientes palabras: Mashup, mashing, mashUp, mash o ...)

Nosotros experimentamos con algunas cadenas de búsqueda y ésta fue la que mejores resultados obtuvo. Esta cadena de búsqueda fue usada en IEEE Xplore y ACM como también usada en las búsquedas manuales en las otras fuentes. El período cubierto fue de 6 años (2006 – 2012). Esta fecha de inicio fue seleccionada debido a que son las referencias de los primeros estudios en los cuales el término “Mashup” empezó a aparecer en el campo de la Ingeniería Web.

Criterios de Inclusión y Exclusión

Cada estudio identificado fue evaluado por investigadores que conducen en mapeo sistemático para decidir si estos deberán o no ser incluidos. Las discrepancias fueron resueltas por consenso. Los estudios que reúnan las siguientes condiciones fueron incluidas.

- Estudios que presentan un método y/o técnica para asistir a los diseñadores en la evaluación de calidad de los Mashups desde la perspectiva del producto, y
- Full papers.
- Los estudios que reúnen al menos una de las siguientes condiciones son excluidos:
- Estudios introductorios para problemas específicos, libros y workshops.
- Reportes duplicados del mismo estudio en diferentes fuentes.
- Estudios cortos con menos de cinco páginas.
- Estudios no escritos en inglés.

Aseguramiento de Calidad

Adicionalmente a los criterios de inclusión/exclusión, se considera crítica la evaluación de la calidad de los estudios primarios. Un cuestionario de tres puntos de la escala de Likert fue usado para proveer una evaluación de la calidad de los estudios seleccionados. El cuestionario con las 3 preguntas son:

- a) ¿El estudio presenta un método o técnica para evaluar localidad de los Mashups desde la perspectiva de calidad del producto?
- b) ¿El estudio ha sido publicado en una revista relevante o conferencia?
- c) ¿El estudio ha sido citado por otros autores?

El puntaje para cada pregunta cerrada será la media aritmética de todos los puntajes individuales de cada revisor. La suma de los puntajes de las 4 preguntas cerradas de cada estudio provee un puntaje final, el cual ha sido usado para excluir los estudios del mapeo sistemático y además es usado para detectar los estudios representativos.

Estrategia de Extracción de datos

La estrategia de extracción fue definida descomponiendo cada pregunta de investigación en criterios más específicos en los cuales un conjunto de opciones posibles fue establecido. La siguiente tabla muestra esta descomposición la cual trata de facilitar la extracción de datos y la categorización de estudios, como se indica en la *Tabla 3.1*.

Con respecto al criterio C1, un estudio puede ser clasificado en una o más características del Estándar SQuARE ISO / IEC 25010 [31]. Nosotros empleamos este estándar porque este propone un modelo de calidad de producto actualizado el cual ha sido definido por consenso entre expertos.

Con respecto al criterio C2, un estudio puede ser clasificado en uno o más estados basados en el proceso de desarrollo de Mashups propuesto en [12]: Selección de componentes, si la calidad del Mashup está evaluada cuando los componentes están siendo seleccionados para crear el Mashup; composición del Mashup, si la calidad del

Mashup es evaluada durante el proceso de composición y uso del Mashup, una vez que la aplicación haya sido completamente construida y esté lista para ser usada.

Pregunta de Investigación	Criterio	Opciones
RQ1: ¿Cuáles/cómo los aspectos de calidad de producto están envueltos en el desarrollo de los Mashups?	C1: Características de la calidad de producto dirigidas en los estudios	Adecuación/Funcionalidad Desempeño / Eficiencia Compatibilidad Usabilidad Confiabilidad Seguridad Mantenibilidad Portabilidad
	C2: Estados en los que se basa el proceso de desarrollo de los Mashups.	Selección de componentes Composición de los Mashups Uso de los Mashups
	C3: Artefactos involucrados	Modelos conceptuales Código fuente Interfaces de usuario final Componentes
RQ2: ¿Cómo los enfoques de calidad de producto han sido definidos y validados?	C4: Tipo de enfoque	Nuevo Extensión
	C5: Tipo de validación	Estudio (Survey) Caso de estudio Experimento No evaluación
	C6: Enfoque en el uso	Industria Academia

Tabla 3-1 Criterios de Extracción. Mapeo Sistemático Calidad Mashups

Con respecto al criterio C3, un estudio puede ser clasificado en uno o más artefactos: modelos conceptuales, si la calidad de los Mashups es evaluada sobre artefactos intermedios que son creados durante el proceso de desarrollo del Mashup (calidad interna); Código fuente, si la calidad del Mashup es evaluada inspeccionando la implementación final (calidad externa), Interfaces de usuario final, si la calidad de los componentes seleccionados es evaluada o considerada antes de mezclarlos.

Con respecto al criterio C4, un estudio puede ser clasificada en una de las siguientes respuestas: Nueva, si esta presenta un enfoque desde sus inicios (por ejemplo una evaluación o técnica específicamente obtenida de la evaluación de la calidad de los Mashups); o existente, si esta presenta una extensión de un enfoque previo (por ejemplo una técnica definida por la evaluación de los servicios Web que han sido aplicados para evaluar los Mashups).

Con respecto al criterio C5, un estudio puede ser clasificado en uno de los siguientes tipos de estrategias que pueden ser llevadas a cabo dependiendo del propósito de la validación y de las condiciones para la investigación empírica [18]: Estudio (Survey), si este prueba una investigación ejecutada en retrospectiva; Caso de estudio, si esta provee un estudio observacional en los cuales los datos son recolectados durante ambientes reales y simulados; Experimento controlado, si este provee una investigación formal, rigurosa y

controlada que está basada en la verificación de una hipótesis; y No validación, si esta no provee ningún estudio empírico relacionado a la calidad de producto para Mashups.

Finalmente, con respecto al criterio C6, un estudio puede ser clasificado de acuerdo al contexto/ambiente en el cual el método/técnica de la evaluación de la calidad ha sido definida o está siendo usada actualmente (contexto industrial y / o contexto académico).

Métodos de Síntesis

Hemos aplicado métodos de síntesis cuantitativos y cualitativos. La síntesis cuantitativa está basada en:

- Contar los estudios primarios que son clasificados en cada respuesta desde nuestro criterio.
- Definir gráficos de burbujas para mostrar las frecuencias de la combinación de resultados desde diferentes sub-preguntas de investigación. Un gráfico de burbujas es básicamente un eje x-y con burbujas en las intersecciones de las categorías. Este es útil para proveer un mapa y dar una rápida visión de un campo de investigación [53].
- Contar el número de estudios encontrados en cada fuente bibliográfica por año.

La síntesis cualitativa está basada en incluir algunos estudios representativos para cada criterio considerando los resultados de cada evaluación de la calidad.

3.2.2. Fase de Conducción

La búsqueda para identificar los estudios primarios las bibliotecas digitales IEEE Xplore y ACM fue realizada el 29 de diciembre de 2012. La aplicación del protocolo de revisión trajo los siguientes resultados:

- La búsqueda bibliográfica en la base de datos identificó 80 publicaciones potencialmente relevantes (46 de IEEE Xplore y 34 de ACM). Luego de aplicar los criterios de inclusión y exclusión documentados anteriormente, fueron finalmente seleccionados (13 de IEEE Xplore y 16 de ACM).
- La revisión manual de la bibliografía de otras fuentes identificaron otras 107 publicaciones potencialmente relevantes. Luego de aplicar los criterios de inclusión y exclusión, se seleccionaron 9 estudios (2 de WWW, 0 de ICWE, 1 de WISE, 1 de iiWAS, 1 de SOSE, 2 de ICSOC y 2 de MASHUPS).

De ahí un total de 38 estudios de investigación fueron seleccionados por nuestros criterios de inclusión/exclusión. Algunos estudios han sido publicados en más de una revista/conferencia. En este caso, nosotros hemos seleccionado solamente la versión más completa del estudio. Los estudios que aparecen en más de una fuente fueron tomadas en cuenta solamente una vez.

3.2.3. Reporte de resultados

Un resumen de los resultados de nuestro estudio está presentado en la tabla 3-2. La lista de estudios incluidos en nuestro mapeo sistemático está disponible en <http://users.dsic.upv.es/~icedillo/mapping/>.

Con respecto al criterio C1 “Características de calidad del producto dirigidas en los estudios”, los resultados indican que la mayoría de atributos de calidad fueron desempeño/eficiencia (63%) y seguridad (47%). La proporción es debido a que la naturaleza del requisito de datos de los Mashups en donde se necesita un acceso rápido y seguro es muy común. Los atributos menos considerados son: la portabilidad (10%), la mantenibilidad (21%) y la compatibilidad (10%). Incluso algunos autores han mencionado que “Los aspectos tales como la mantenibilidad o escalabilidad juegan un rol menor porque generalmente un Mashup es necesario solamente por un corto período de tiempo” [13]. Sin embargo la adecuación/funcionalidad (42%) y confiabilidad (42%) recibieron menos consideración que la esperada, la usabilidad solamente se abordó en el 23% de los estudios a pesar de que usualmente es una de las características más deseables y relevantes en el dominio Web.

Criterio	Posibles Respuestas	#	%
		Estudios	Porcentaje
C1: Características de la calidad de producto dirigidas en los estudios	Adecuación / Funcional.	16	42,11
	Desempeño / Eficiencia	24	63,16
	Compatibilidad	4	10,53
	Usabilidad	9	23,68
	Confiabilidad	16	42,11
	Seguridad	18	47,37
	Mantenibilidad	8	21,05
	Portabilidad	4	10,53
C2: Estados en los que se basa el proceso de desarrollo de los Mashups.	Selección de Componentes	14	36,84
	Composición de los Mashups	21	55,26
	Uso de los Mashups	16	42,11
C3: Artefactos involucrados	Modelos conceptuales	8	21,05
	Código fuente	13	34,21
	Interfaces de usuario fina	9	23,68
	Componentes	15	39,47
C4: Tipo de enfoque	Nuevo	29	76,32
	Extensión	9	23,68
C5: Tipo de validación	Estudio (Survey)	1	2,63
	Caso de estudio	18	47,37
	Experimento	9	23,68
	No Evaluación	11	28,95
C6: Enfoque en el uso	Industria	15	39,47
	Academia	33	86,84

Tabla 3-2 *Mapeo Sistemático. Resultados según criterios de extracción*

Con respecto al criterio C2 “Estados en los que se basa el proceso de desarrollo de los Mashups”, los resultados indican que la mayoría de estudios están de acuerdo en direccionar los aspectos de calidad cuando el Mashup está yendo a ser compuesto (55%). Estos estudios tratan de mejorar el proceso de composición para así obtener un Mashup de mejor calidad. Por ejemplo, en [7], se presentó una técnica de composición basada en pipelines para así mejorar la adecuación / funcionalidad y desempeño del Mashup obtenido.

El 42% de los estudios se centraron en cómo mejorar la calidad cuando el Mashup esté completado. Algunos de estos estudios tales como [49], tratan de ofrecer recomendaciones a estados previos del proceso de desarrollo, donde otros estudios tales como [33], se han preocupado de evaluar la calidad en el uso para reportar problemas. Finalmente, pocos estudios consideraron que la selección de componentes está dirigida por aspectos de calidad (37%). Nosotros argumentamos que más estudios tales como [12] son necesarios para mejorar los métodos o guías para seleccionar los componentes a fin de obtener un mejor Mashup.

Con respecto al criterio C3 “Artefactos envueltos”, los resultados indican que la mayoría de los estudios dirigen los problemas de calidad a los componentes seleccionados para ser mezclados (40%). Un ejemplo puede ser encontrado en [12] donde los componentes son previamente puntuados para así ayudar a la evaluación del Mashup obtenido. El 34% de los estudios abordan la calidad en el código fuente del Mashup. Un ejemplo puede ser encontrado en [57], en donde un algoritmo analiza el código fuente para mejorar aspectos de calidad. Existen pocos estudios que consideran modelos conceptuales (10%) que definen la composición del componente para mejorar problemas de calidad. Nosotros encontramos un ejemplo de este tipo de estudios en [8], en donde los modelos de orquestación y negocios son analizados y mejoran la aceptación de Mashup. Finalmente el 23% de los estudios analizan la interacción del Mashup para descubrir deficiencias, tales como el [37], donde áreas para definir métricas fueron exploradas.

Con respecto al criterio C4, “Tipo de enfoque”, los resultados indican que la mayoría de estudios (76%) presentan nuevos enfoques para tratar aspectos de calidad sobre Mashups. Algunos de estos estudios tales como [42], proponen ideas inspirándose en otros dominios. Nosotros argumentamos que estos descubrimientos muestran un acuerdo entre los autores en la importancia de considerar a los Mashups no solamente como aplicaciones Web simples.

Con respecto al criterio C5 “tipo de validación”, los resultados indican que la mayoría de los estudios han presentado Casos de estudio para validar sus enfoques (47%). Este es un resultado que mejora la situación descrita en una revisión sistemática presentada en [21], la misma que muestra una falta de estudios rigurosos en la investigación de la Ingeniería Web. Un ejemplo de un caso de estudio puede ser encontrado en [66]. Sin embargo, pocos Experimentos han sido conducidos (23%). Los experimentos deberían ser más empleados ya que ellos proveen un alto nivel de control y un enfoque de evaluación útil en un modo más riguroso. Un ejemplo de un experimento puede ser encontrado en [55]. Finalmente, estudios (surveys) son los menos preferidos para las investigaciones (3%) y el resto de estudios (29%) no proveen ninguna clase de validación ya que solo describen pruebas de conceptos.

Con respecto al criterio C6 “Enfoque de uso”, los resultados indican que la mayoría de los estudios (87%) han sido ejecutados desde la investigación académica, por ejemplo en [49]. Sin embargo es importante notar que un 39% de los estudios fueron ejecutados desde el punto de

vista de la industria. Estos estudios, tales como [59], estaban especialmente interesados en direccionar los problemas de calidad, los cuales fueron detectados como relevantes por los participantes.

Es necesario mencionar que el análisis del número de estudios de investigación en calidad para Mashups mostrados ha sido de gran interés desde el 2007. La figura 3-2, muestra el número de publicaciones seleccionadas por año y fuente. Nosotros creemos que este interés en crecimiento muestra la importancia de conducir estudios basados en esta área.

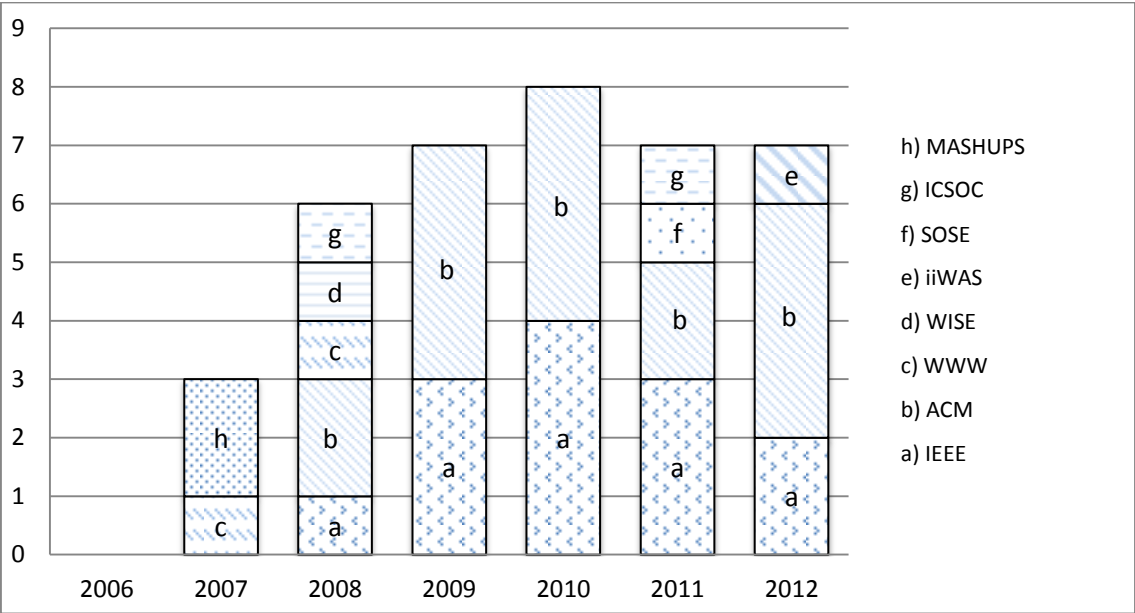


Figura 3-2 Publicaciones sobre Calidad en Mashups.

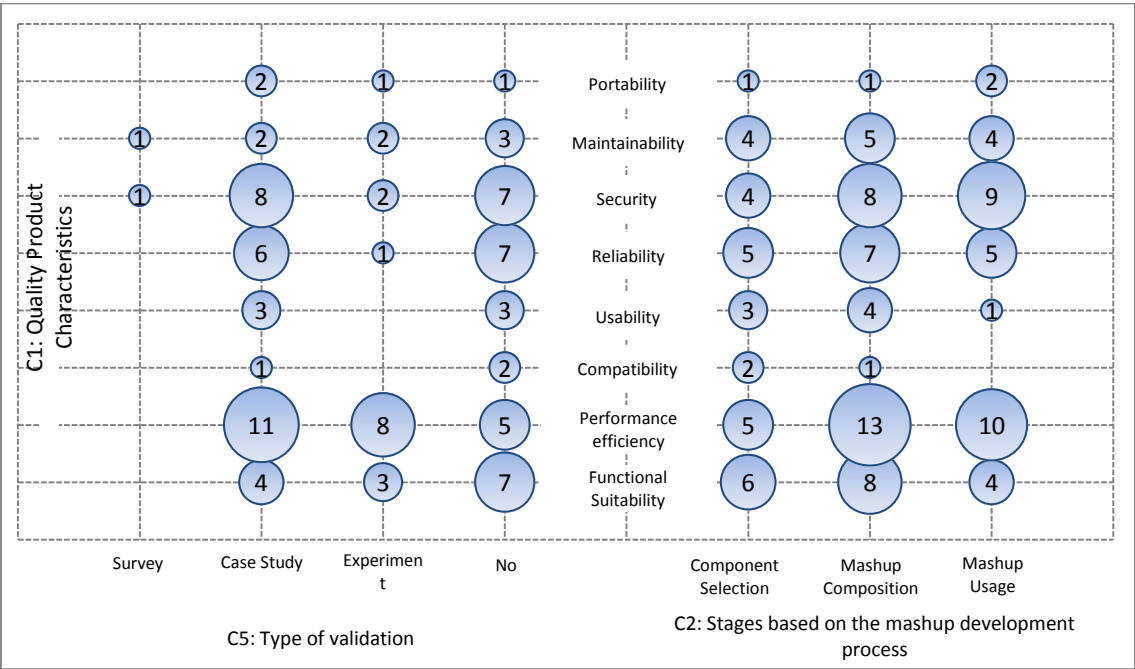


Figura 3-3 Comparación entre criterios de extracción C1, C2, C5

El criterio fue combinado para establecer un mapeo con el fin de proveer una visión de los aspectos de calidad para Mashups. Este mapeo nos permite obtener más información sobre cómo los resultados de cada criterio están relacionados a los otros, y cuáles son los posibles aspectos de investigación que aún no han sido abordados. A continuación se presentan los gráficos de burbujas los cuales están relacionados a comparar los criterios C1 “características de calidad direccionadas” contra el criterio C2 “estados” y C4 “tipo de validación”.

También se ha incluido otro gráfico que compara el criterio C1: características de calidad del producto con c4: el tipo de enfoque de los estudios y c6: el enfoque en el uso, incluyéndose también en el mismo la comparación de C3: los artefactos involucrados con C6: Enfoque en el uso y con C4: Tipo de enfoque.

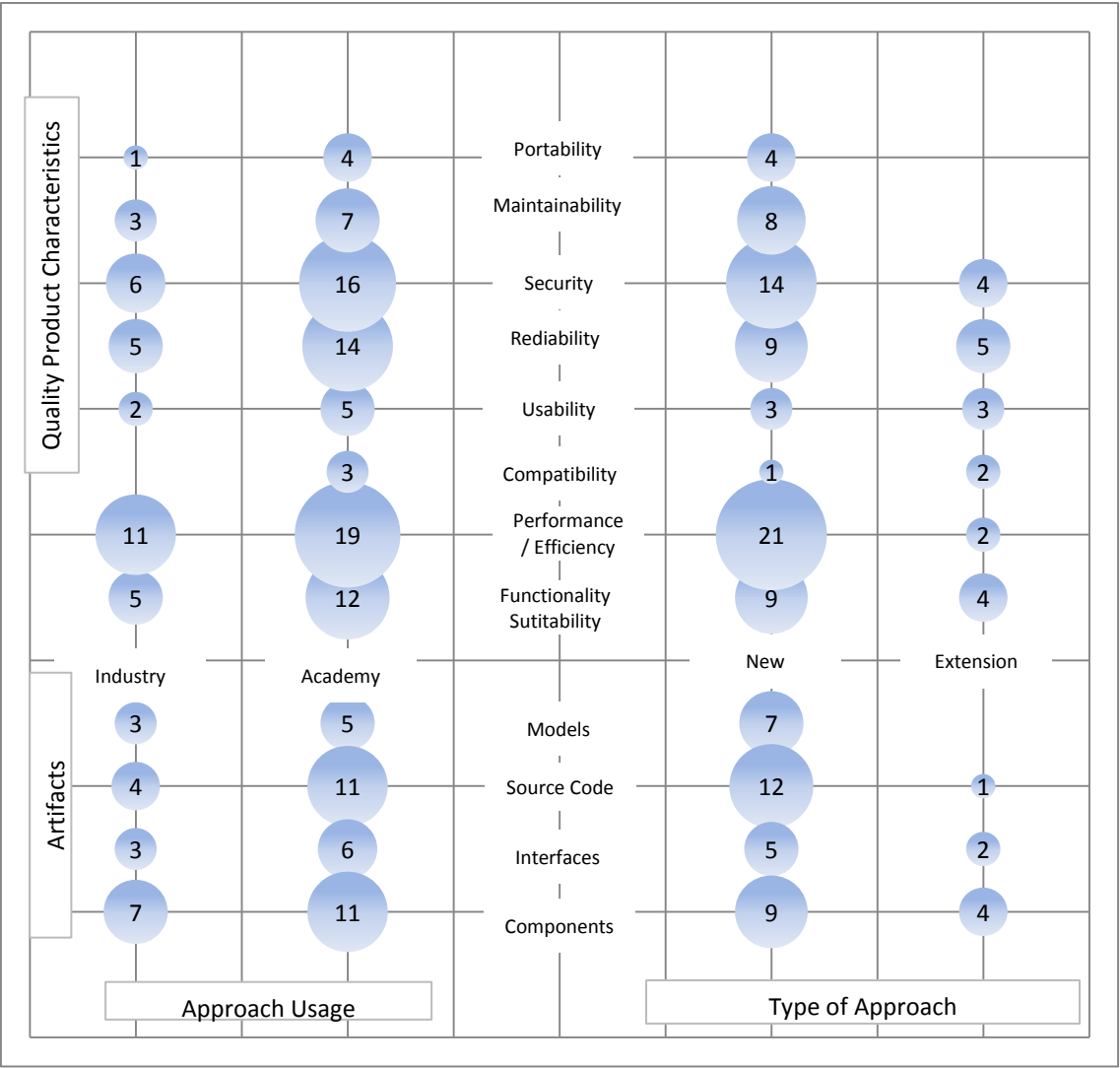


Figura 3-4 Criterios de extracción Artefactos-Características de Calidad de Producto--Tipos de enfoque

Hemos incluido el gráfico que compara el criterio C1: características de calidad del producto con C2 “estados” y C5: “tipo de validación”, incluyéndose también en el mismo la comparación de C3: “artefactos involucrados” con C2: “estados” y con C5: “Tipo de validación”.

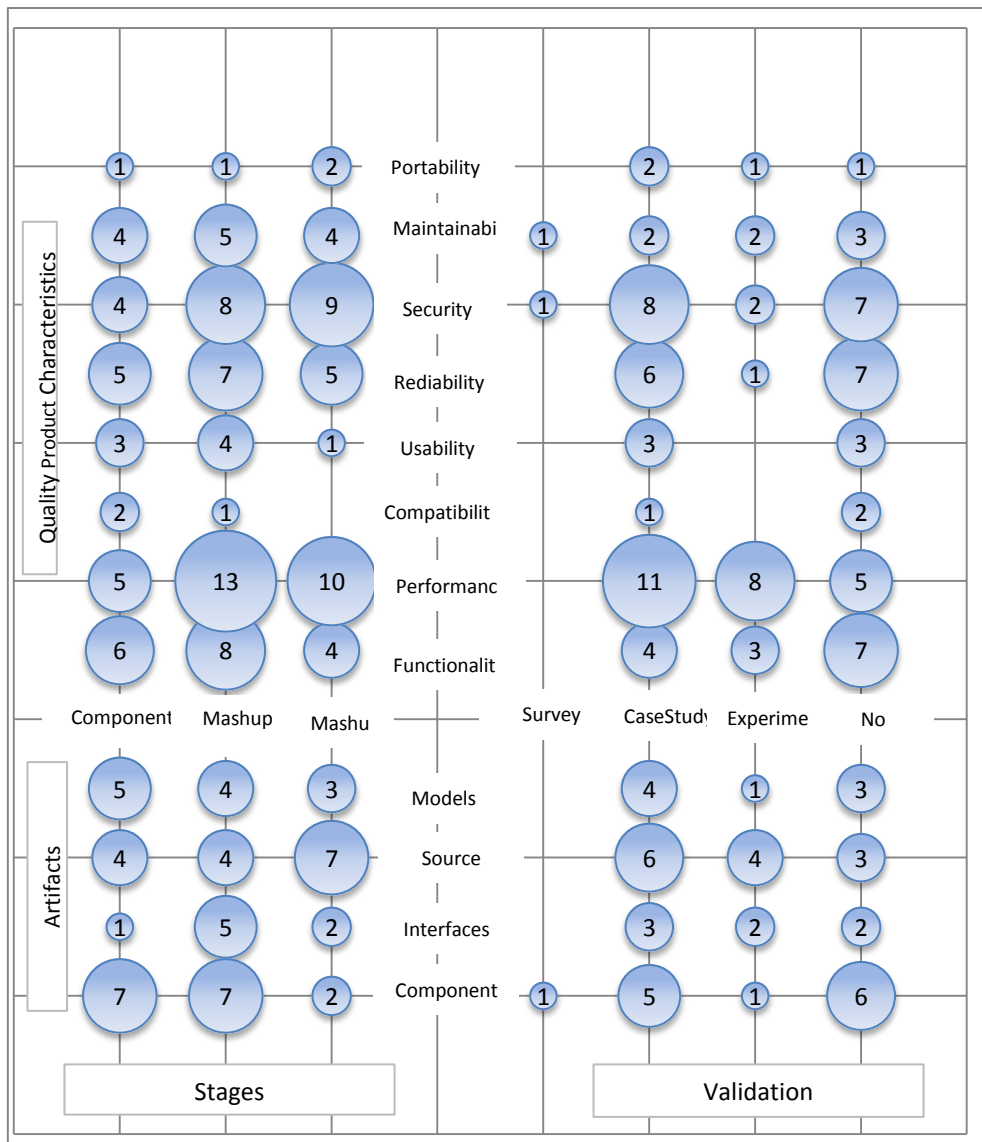


Figura 3-5 Criterios de extracción Artefactos-Características de Calidad de Producto—Estados del ciclo de vida-Validación

El mapeo resultante confirma que la mayoría estudios de características de calidad se da en el estado de Composición del Mashup. Sin embargo, este muestra lagunas de investigación tales como características de calidad que no han sido validadas a través de experimentos controlados (por ejemplo: seguridad, usabilidad, confiabilidad).

Dificultades en la Validación

Las principales dificultades de este estudio son: el sesgo en el ámbito de nuestras preguntas de investigación, publicación y selección, la imprecisión en la extracción de datos y errores en la clasificación.

El ámbito de nuestra pregunta de investigación fue limitada a la calidad del producto de los Mashups. Sin embargo, nosotros observamos durante la conducción de este mapeo que la calidad del proceso de desarrollo es una extensión interesante la cual será explorada como trabajo futuro.

Los sesgos de publicación se refieren al problema de que los resultados positivos son más deseables de publicación que los negativos [21]. Nosotros estamos conscientes de que esta es una limitación inherente a nuestras fuentes bibliográficas. Con respecto al sesgo en la selección de publicaciones, nosotros escogimos las fuentes donde los estudios sobre el desarrollo de Mashups son normalmente publicados y comparamos los estudios obtenidos contra una pequeña muestra la cual fue previamente identificada como estudios relevantes. Sin embargo, nosotros no consideramos otras fuentes bibliográficas tales como Springer Link o ScienceDirect lo que puede haber afectado a la completitud de nuestro mapeo sistemático. Además, nuestra búsqueda bibliográfica fue conducida al final del año 2012, algunos estudios no estuvieron aún indexados en ese momento y pudieron no haber sido considerados.

Finalmente, nosotros intentamos aliviar la imprecisión en la extracción de los datos y la falta de una adecuada clasificación por medio de tres revisores y la resolución de discrepancias por consenso.

3.3. Mapeo Sistemático sobre la Composicionalidad de los Mashups

Además del mapeo sistemático sobre la calidad de producto en Mashups, presentado en la sección 3.2, en esta sección se presentan los resultados de otro mapeo sistemático, conducido para identificar los aspectos de composicionalidad relevantes para Mashups. Este mapeo sistemático se ha enfocado a la pregunta de investigación: “QUÉ CRITERIOS DE COMPOSICIÓN SON EMPLEADOS POR LOS INTEGRADORES AL MOMENTO DE CREAR MASHUPS”. Se explicará como el mapeo sistemático ha sido conducido y los resultados que se han obtenido del mismo. Luego se comentarán los principales hallazgos, las implicaciones para los investigadores y las limitaciones del estudio.

Los Mashups al ser aplicaciones Web construidas en base a recursos provistos por terceros, mantiene características muy particulares a su esencia composicional, estas características son las que hemos cubierto en esta sección.

A fin de lograr una clasificación y revisión de estas características se han revisado estudios que tratan sobre aspectos relacionados a los Mashups, por ejemplo en [9], se presenta una revisión amplia que ayuda a enmarcar los temas de la investigación de los Mashups en general, desarrollando una taxonomía de varios aspectos relacionados a los mismos, los autores realizaron una clasificación en seis categorías de este tipo de aplicaciones basada en 60 publicaciones: control de acceso y comunicación cruzada, integración de los Mashups, agentes, marcos de trabajo, programación del lado del usuario y Mashups empresariales; y ha servido como una de las principales fuentes bibliográficas para el presente estudio.

Otro importante aporte encontrado, fue el realizado por [52], proyecto que tiene como objetivo proveer una plataforma distribuida para integración dirigida por eventos entre servicios, que se da a nivel de Internet y que habla realiza una clasificación de los Mashups con el fin de enmarcar y estructurar sus criterios de selección y evaluación de las herramientas existentes así como también para refinar los requerimientos para diseñar la herramienta.

Aquí, se ha realizado una exhaustiva investigación acerca de los aspectos de composicionalidad más relevantes en el ámbito de los Mashups. Para ello se ha empleado un método que permita ubicar toda la información actual mediante un Mapeo Sistemático.

Método de Investigación

Se ha elaborado un mapeo sistemático considerando las guías provistas en [36] [53]. Un mapeo sistemático es la categorización y resumen de la información existente sobre una pregunta de investigación en una manera imparcial.

El mapeo sistemático se ejecutó entres estados: Planificación, Conducción y Reporte de resultados.

3.3.1. Fase de Planificación

Al igual que en la sección 3.2, se han realizado los mismos pasos en la fase de planificación de este mapeo sistemático, los mismos que son explicados a continuación:

Pregunta de Investigación

El objetivo de este mapeo sistemático es examinar la situación actual de los aspectos de composicionalidad de los Mashups y para ello se ha establecido la siguiente pregunta de investigación: “QUÉ CRITERIOS DE COMPOSICIÓN SON EMPLEADOS POR LOS INTEGRADORES AL MOMENTO DE CREAR MASHUPS”. Esto nos permitirá categorizar y resumir el conocimiento actual referente a los aspectos de composicionalidad en los Mashups e identificar las carencias en la investigación actual para de esta manera sugerir áreas para posible investigación futura.

Adicionalmente a esto y con el fin de responder la pregunta de investigación se han identificado sub-preguntas más detalladas. La siguiente tabla muestra las sub-preguntas y su motivación:

RQ1: ¿Cómo está dirigida la composición en el desarrollo de los Mashups?

RQ2: ¿Cómo está direccionada la investigación en los estudios referentes a los Mashups?

Estas preguntas de investigación nos permitirán conocer las guías y las maneras de componer Mashups, si se están aplicando técnicas de composicionalidad, saber éstas en qué se basan y conocer si existen herramientas o metodologías que apoyen en el proceso de composición al desarrollador de los Mashups. Así también nos permitirán conocer cómo se está llevando a cabo la investigación en lo relacionado a los Mashups.

Estrategia de Búsqueda

Las bibliotecas digitales usadas para la búsqueda automática de los estudios primarios fueron:

- ACM Digital Library.
- IEEE Xplore Digital Library.

La búsqueda manual fue realizada en conferencias, workshops y revistas, esta búsqueda fue realizada en:

- Conferencias y workshops:
 - International World Wide Web conference – WWW(2006-2012).
 - International Conference on Web Engineering – ICWE (2006-2012).

- International Conference on Web Information Systems Engineering – WISE (2006-2012)
- International Conference on Information Integration and Web-based Applications & Services iiWAS (2006-2012).
- IEEE International Symposium on Service Oriented System Engineering SOSE (2006-2012).
- Workshop on Web APIs and Service Mashups – MASHUPS (2006-2012).
- International Conference on Service Oriented Computing – ICSOC (2006-2012).
- Revistas y libros:
 - Journal of Web Engineering (JWE).
 - ACM Transactions on the Web.
 - Foundations of Popfly: Rapid Mashup Development (Book for Professional by Professionals).

Para ejecutar la búsqueda automática de las librerías digitales seleccionadas, se ha utilizado una cadena de búsqueda que permita cubrir los conceptos y los estudios para responder la pregunta de investigación, dicha cadena de búsqueda se indica en la siguiente tabla:

Concepto	Sub-Cadena	Conector	Términos alternativos
Mashup	Mash*	AND	Esto incluye: Mash, Mashup, mashing, mash-Up, etc.
Composición	Compos*		Esto incluye: compose, composition, composing, compositionality, etc.
Cadena de búsqueda definitiva	MASH* AND COMPOS*		

Tabla 3-3 **Formación del String de búsqueda del Mapeo Sistemático**

La búsqueda fue realizada aplicando la cadena antes descrita a los metadatos del título, del resumen y a las palabras claves de cada artículo para todas las fuentes (la sintaxis de la cadena de búsqueda se adaptó a cada librería digital).

Estos términos de búsqueda son además tomados en cuenta para las otras fuentes que fueron manualmente inspeccionadas para obtener así una búsqueda consistente.

El período de revisión incluye estudios publicados desde el año 2006 hasta el 2012. Esta fecha de inicio fue seleccionada porque en el año 2006 el término Mashup se acuñó en los primeros trabajos referentes a este dominio.

Selección de Estudios Primarios

Cada uno de los estudios resultado de las búsquedas automáticas y manuales fue evaluado por tres personas (el autor de esta tesis y sus dos supervisores) con el fin de decidir si se debería o no incluir el mismo considerando su título, resumen o palabras clave. Las discrepancias en la selección fueron solventadas por medio de un consenso entre las mismas luego de analizar rápidamente el trabajo. Los estudios que reúnen al menos uno de los siguientes criterios de inclusión fueron adoptados:

- Estudios que presenten aspectos de composicionalidad en Mashups.

Los estudios que reúnan uno de los siguientes criterios de exclusión fueron descartados:

- Estudios introductorios para ediciones especiales, libros y workshops.
- Reportes duplicados del mismo estudio en diferentes fuentes.
- Estudios cortos con menos de cinco páginas.
- Trabajos que no han sido presentados en inglés.

Aseguramiento de la Calidad

Un cuestionario en la escala de Likert fue diseñado para proveer un aseguramiento de la calidad de los estudios seleccionados. El cuestionario contiene dos preguntas cerradas subjetivas y dos preguntas cerradas objetivas.

Las preguntas subjetivas fueron:

- a. El estudio presente aspectos de composicionalidad en Mashups.

Las posibles respuestas a esas preguntas son: “Totalmente de acuerdo” (+1), “Parcialmente de acuerdo” (0) y “Totalmente en desacuerdo (-1)”.

Las preguntas objetivas son las siguientes:

- b. El estudio ha sido publicado en una revista o conferencia relevante.

Las posibles respuestas a esta pregunta son: “Muy relevante” (+1) , “Relevante” (0), “No muy relevante” (-1).

Esta pregunta fue además puntuada considerando el orden de relevancia provista por las librerías digitales, el ranking de conferencias CORE (A, B, C) y los reportes de citación de revistas (JCR).

- c. El trabajo ha sido citado por otros autores.

Las posibles respuestas son: Si el estudio ha sido citado por más de cinco autores (+1); si el estudio ha sido citado de 1 a 5 veces por autores diferentes (0) y si el estudio no ha sido citado (-1).

Esta pregunta ha sido puntuada considerando el conteo de citaciones hecho por Google Académico. Es importante destacar que el puntaje mínimo para publicaciones recientes es considerado como 0 (parcialmente) para de esta forma no penalizarlo.

El puntaje de cada pregunta cerrada será la media aritmética de todos los puntajes individuales de cada revisor. La suma de las cuatro preguntas cerradas de cada estudio provee un puntaje final (un número entero entre -4 y 4). Estos puntajes no fueron utilizados para excluir estudios del mapeo sistemático sino más bien para detectar estudios representativos para la discusión de la pregunta y sub-preguntas de investigación.

Estrategia de Extracción de Datos

La estrategia de extracción de datos que fue empleada está basada en la provisión de un conjunto de posibilidades que permitan resolver las sub-preguntas de investigación para

así poder responder la pregunta de investigación total que fue definida. Esta estrategia asegura la aplicación de los mismos criterios de extracción para todos los estudios seleccionados y facilita su clasificación.

Pregunta de Investigación	Criterio	Opciones
RQ1: ¿Cómo está dirigida la composición en el desarrollo de los Mashups?	C1: Integración de componentes	Homogéneos Heterogéneos
	C2: Identificación y selección de componentes.	Automática Manual
	C3: Composición de los componentes	Lado del servidor Lado del cliente
	C4: Contexto de la aplicación	Empresa Cliente Negocios
	C5: Nivel de la composición	Presentación Datos Proceso del Negocio
	C6: Mecanismos de composición	Agregación Integración
	C7: Tecnologías involucradas	Fundación Presentación Interactividad Servicios Web Datos
	C8: Tipo de herramienta empleada	Gráfica / Desarrollada Lenguaje / No desarrollada
	C9: Interacciones de los componentes	Orquestación Coreografía
RQ2: ¿Cómo está enfocada la investigación en los estudios referentes a los Mashups?	C10: Fases en las cuales los estudios están basados	Selección de componentes Composición Uso
	C11: Artefactos usados	Modelos conceptuales Código fuente Interfaces de usuario Componentes
	C12: Tipo de validación	Estudio (Survey) Caso de estudio Experimento No evaluación
	C13: Entorno de investigación	Industria Academia
	C14: Tipo de validación	Estudios / Inspecciones Caso de estudio Experimentos Otros
	C15: Entorno de investigación	Industria Academia
	C16: Ambiente de uso	Aplicaciones Móviles Aplicaciones Web Otros
	C17: Metodología	Nueva Extensión

Tabla 3-4

Criterios de extracción. Mapeo sistemático sobre composicionalidad

En la tabla 3-4 se describen los criterios de extracción a emplearse en el presente estudio.

Métodos de Síntesis

Para este mapeo sistemático, se han aplicado tanto métodos de síntesis cualitativos como cuantitativos.

Los métodos de síntesis cuantitativos están basados en:

- Contar los estudios primarios que están clasificados en cada respuesta desde nuestras sub-preguntas de investigación.
- Definir gráficos en burbuja con el fin de reportar las frecuencias de combinar los resultados desde diferentes sub-preguntas de investigación. Un gráfico de burbuja es básicamente aquel con las variables x e y con burbujas en las intersecciones de las categorías. Este método de síntesis es útil para proveer un mapa y dar una rápida visión de un campo de investigación [9].
- Contar el número de estudios encontrados en cada fuente bibliográfica por año.

La síntesis cualitativa está basada en:

- Incluir algunos estudios representativos para cada sub-pregunta de investigación considerando los resultados desde los aseguradores de calidad.

3.3.2. Fase de Conducción

Con la aplicación del protocolo de revisión se obtuvieron resultados preliminares. Los resultados fueron seleccionados de acuerdo con el criterio de inclusión.

Presentamos una tabla que puede resumir los resultados del estudio de conducción, la fuente de las búsquedas automáticas, los estudios potenciales y los estudios seleccionados.

En este estado nosotros pudimos encontrar algunos estudios que posiblemente fueron publicados en más de una revista o conferencia y en este caso, nosotros seleccionamos solamente la versión más completa del estudio. Así mismo en publicaciones iguales, consideramos el orden de búsqueda: IEEEExplore, ACM, Springer Link y Science Direct.

La aplicación del protocolo de revisión trajo los estudios preliminares mostrados en la tabla 3-6.

Un total de 97 publicaciones fueron seleccionadas de acuerdo a los criterios de inclusión/exclusión. En la extracción de los trabajos finales, se contemplaron los siguientes hechos:

- Algunos estudios han sido publicados en más de una conferencia / revista. En este caso se seleccionaron solamente las versiones más completas del estudio.
- Algunos estudios aparecían en más de una fuente. En este caso se tomaron en cuenta solamente una vez de acuerdo al orden de fuentes presentadas anteriormente.

Fuente		Estudios potenciales	Estudios seleccionados
Búsqueda Automática	IEEEExplore (IEEE)	147	52
	ACM DL(ACM)	90	22
	Total	237	74
Búsqueda Manual	WWW Conference	17	2
	ICWE Conference	30	3
	WISE Conference	11	6
	iiWAS Conference	17	3
	ICSOC Conference	17	5
	MASHUPS	10	4
	Total	102	23
	Total Final	339	97

Tabla 3-5 Resultados fase de conducción

3.3.3. Reporte de Resultados

El promedio de resultados, los cuales están basados en el conteo de los estudios primarios que son clasificados en cada una de las respuestas a las sub-preguntas, son presentados en la tabla 3-7. Los lectores que deseen revisar la lista completa de estudios seleccionados incluida en este mapeo sistemático se pueden referir al Apéndice A.

Código	Criterio de Extracción	Respuestas Posibles	#	%
			Estudios	Porcentaje
RQ1	¿Cómo esta direccionada la composición en el desarrollo de los Web Mashups?			
C1	Integración de componentes	Homogéneos	6	6,19
		Heterogéneos	94	96,91
C2	Identificación y selección de componentes	Automática	40	41,24
		Manual	62	63,92
C3	Composición de componentes	Lado-Servidor	31	31,96
		Lado-Cliente	31	31,96
C4	Contexto de aplicación	Empresa	36	37,11
		Consumo/cliente	58	59,79
		Negocios	25	25,77
C5	Nivel de Mashups	Presentación	42	43,30
		Datos	61	62,89
		Flujo de procesos	81	83,51
C6	Mecanismos de composición.	Agregación	39	40,21
		Integración	28	28,87
C7	Tecnologías involucradas	Fundación	9	9,28
		Presentación	31	31,96
		Interactividad	26	26,80
		Servicios Web	75	77,32
		Datos	53	54,64
C8	Tipo de herramienta empleada	Cliente (Gráfica)	39	40,21
		Desarrollador (Lenguaje)	51	52,58
C9	Interacción de componentes	Orquestación	26	26,80

Código	Criterio de Extracción	Respuestas Posibles	#	%
		Coreografía	4	4,12
RQ2	¿Cómo está dirigida la investigación en los estudios de los Web Mashups?			
C12	Fases en las cuales los estudios están basados	Selección de componentes	37	28,91
		Fase de composición	106	82,81
		Fase de uso	105	82,03
C13	Artefactos usados	Modelos	22	17,19
		Código fuente	13	10,16
		Otros	4	3,13
		Componentes	64	50,00
C14	Tipos de validación	Estudio / Encuesta	3	2,34
		Caso de Estudio	38	29,69
		Experimento	21	16,41
		Otros	42	32,81
C15	Ámbito de uso	Industria	49	38,28
		Academia	76	59,38
C16	Ambiente de uso	Aplicación Móvil	10	7,81
		Aplicación Web	90	70,31
		Otros	8	6,25
C17	Metodología	Nueva	83	64,84
		Existente	14	10,94

Tabla 3-6 Fase de reporte de resultados

Los criterios de extracción no inclusivos en los cuales los estudios pueden ser clasificados en más de una respuesta no necesariamente sumarán el 100%, pudiendo estos exceder la totalidad de trabajos. Así también existen criterios de extracción para los cuales el estudio no se enmarca en sus opciones asumiendo en este caso que no se aborda el tema relacionado con el criterio.

Con respecto al criterio C1 “Integración de componentes”, los recursos indican que los Mashups en su gran mayoría son compuestos a través de fuentes heterogéneas de datos (96,91%). Esta proporción es debido a que la naturaleza de los mismos trata de unir recursos provistos por terceros, ya sean recursos Web reusables de muchos tipos, pudiendo ser en forma de APIS, Web Widgets y fuentes de datos Web [2], que al estar disponibles facilitan el desarrollo de este tipo de aplicaciones por medio de su composición.

El criterio C2 “Identificación y selección de componentes”, se ha clasificado en automática o manual, cuando hablamos de automática, podemos ver varias herramientas y técnicas que ayudan en la identificación y selección de los componentes necesarios para formar la aplicación Web, al hablar de métodos automáticos de identificación tenemos por ejemplo formas semánticas de descubrimiento de componentes [15] [40] [49], otro ejemplo es a través de redes bayesianas [65] o por medio de Aprendizaje de Máquina [28] [70], además el proceso puede ser semi-automático, esto ocurre cuando la selección puede ser asistida o también puede basarse en recomendaciones no concluyentes las mismas que el usuario puede admitir o descartar. Estos procesos automáticos están tomando fuerza lo que se corrobora mediante el porcentaje de estudios que hablan de métodos que permiten automatizar el proceso de selección (41,24%).

El criterio C3 “Composición de componentes”, trata sobre el lugar en el cual se llevará a cabo la composición de los recursos, pudiendo ser de lado del cliente o del lado del servidor, se ve un equilibrio entre estudios de composición (31,96%), aquellos que no mencionan el lugar en el que se realizará la misma es porque tratan otros temas no relacionados con esto, sin embargo se ve que no hay una tendencia del lado de la arquitectura puesto que se trata de diferente tipo de aplicaciones en las cuales muchas veces es necesaria la composición en el browser [27][34] y otras en las que el mashup emana directamente desde el servidor [15][24] por motivos de seguridad o políticas propias del desarrollador de la aplicación o del proveedor de los recursos.

El criterio C4 “Contexto de aplicación”, permite la clasificación de los Mashups ya sea si estos serán Mashups Empresariales (37,11%), Mashups de Consumo o Cliente Final (59,79%) o por último Mashups orientados a negocios (25,77%), la distinción entre los Mashups Empresariales y los Mashups de Negocios es que un Mashup Empresarial se utiliza dentro del ámbito de la empresa para solventar una necesidad situacional propia de la misma, esta usualmente combina fuentes internas y externas. Un Mashup de negocios es una combinación que pone el resultado disponible para una aplicación de negocios como por ejemplo los de búsqueda de viajes, que permiten al usuario escoger alternativas o ver información pertinente a negocios puntuales [64]. Y por último los Mashups de consumidor, combinan datos desde diferentes sitios Web y usan una interface gráfica simple unificada para mostrar la información combinada. Un ejemplo de los Mashups de consumidor podría ser una aplicación Web que integra fotos digitales desde flickr.com y las muestra sobre Google Maps usando las imágenes [64].

El criterio C5, habla sobre el nivel de los Mashups, si estos están compuestos a nivel de Presentación [4] [70], Datos [41] o Flujo de procesos, los estudios [64], hablan ya de Mashups de Datos como un tipo más de Mashups, aquí lo hemos incluido bajo el criterio del Nivel de Mashups, por cuanto es interesante ver en una arquitectura de capas en donde se da mayormente la composición, obteniendo como resultado a nivel de presentación (43,30%), de datos (62,89%) y a nivel de procesos (83,51%). Este resultado por tanto revela que existe una cantidad de Mashups en los cuales los servicios Web u otras fuentes que proveen datos o procesos se imponen a la hora de construir este tipo de aplicaciones Web.

El criterio C6, trata sobre los “mecanismos de composición” [16], los mismos que pueden ser mediante agregación (40,21%), esto cuando no se requieren habilidades de programación, es decir cuando el usuario final puede componer sin necesidad de recurrir a modificaciones del código fuente o mediante integración (28,87%), en las cuales se requiere un compositor de Mashups con conocimientos de programación, en donde sea necesaria la intervención en el código fuente para la composición de los componentes. Como vemos la suma de los dos porcentajes no llega a cubrir el total de los estudios analizados en este trabajo, y el motivo es que muchas veces se plantean estudios en los cuales no se especifica para qué tipo de compositor se realiza el estudio, esto puede darse porque no necesariamente se está abarcando el estado de composición, sino también los estudios están dirigidos a otros estados del ciclo de vida del Mashup pudiendo ser por ejemplo la selección de componentes o evaluaciones de calidad en productos terminados, en las cuales no se menciona quien ha construido la aplicación [45][65].

El criterio C7, “Tecnologías Involucradas” tiene que ver con el tipo de recursos con los cuales es construido el Mashup, estos pueden ser según [23] fundación o cimentación de la aplicación (9,28%), presentación (31,96%), interactividad (26,80%), Servicios Web – API (77,32%) o Datos (54,64%). En cuanto a la fundación se habla del tipo de tecnologías que sirven de cimentación a todas las demás, pudiendo estas ser HTTP o basadas en el Browser, en los estudios se ha puntuado únicamente aquellos papers que explícitamente mencionan la tecnología, sin embargo cabe destacar que en el caso de “fundación”, los artículos en su mayoría se verían implícitamente enmarcados en cualquiera de las dos opciones. Así también en cuanto a las tecnologías de presentación encontramos HTML/XHTML, CSS siendo principalmente aquellas que nos permiten lograr la visualización de la presentación de la aplicación. Además tenemos las tecnologías relacionadas con la interactividad, estas pueden ser Java Script o Ajax. Las tecnologías asociadas a los servicios Web son XMLHttpRequest, XML-RPC, SOAP y REST y por último las tecnologías de datos pudiendo ser estas XML, RSS/Atom, JSON o KML.

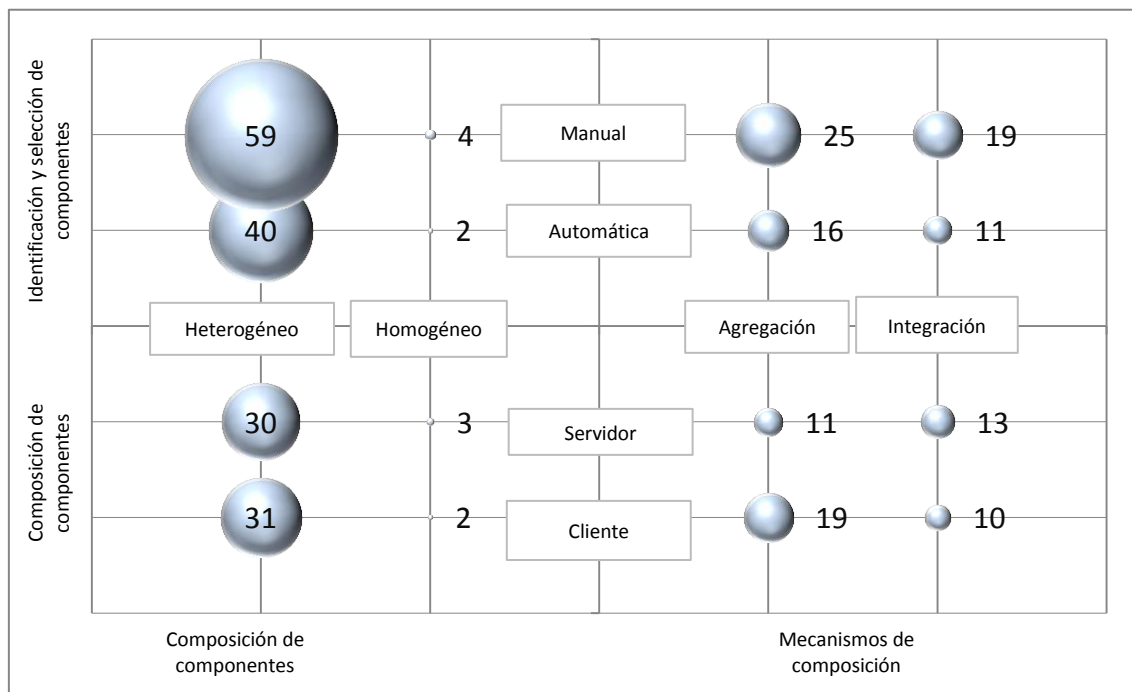


Figura 3-6 Relaciones entre criterios de extracción. Mapeo Sistemático Composicionalidad

El criterio C8 trata sobre el “Tipo de Herramienta Empleada”, en este caso se clasificarán los estudios dependiendo si existe o no una herramienta implicada en el proceso de construcción [26][35] y si esta herramienta es gráfica (40,21%) o trata de un lenguaje de programación u otro tipo de aporte (52,58%), ciertos estudios no se enmarcan en ninguna de las dos opciones de este criterio de extracción ya que su objetivo de estudio puede ser el realizar una evaluación por ejemplo.

Por último dentro de la sub-pregunta de investigación 1, tenemos la “Interacción de Componentes”, la misma que puede ser mediante Orquestación [61] o mediante Coreografía [46].

La sub-pregunta de investigación RQ2, tiene criterios de extracción idénticos a los discutidos en el mapeo sistemático sobre calidad en Mashups del apartado anterior. Se han

considerado por cuanto es necesario saber la dirección que sigue la investigación en cuanto a la composicionalidad de las aplicaciones Web Mashups.

Las siguientes sub-secciones presentan el análisis de los resultados de cada una de las sub - preguntas de investigación y sus criterios de extracción, el mapa creado por combinar diferentes sub – preguntas y cómo han sido cubiertos los temas composicionalidad en los Mashups. Para ello se muestran las gráficas de las relaciones entre los criterios de extracción con el fin de conocer las relaciones entre los mismos.

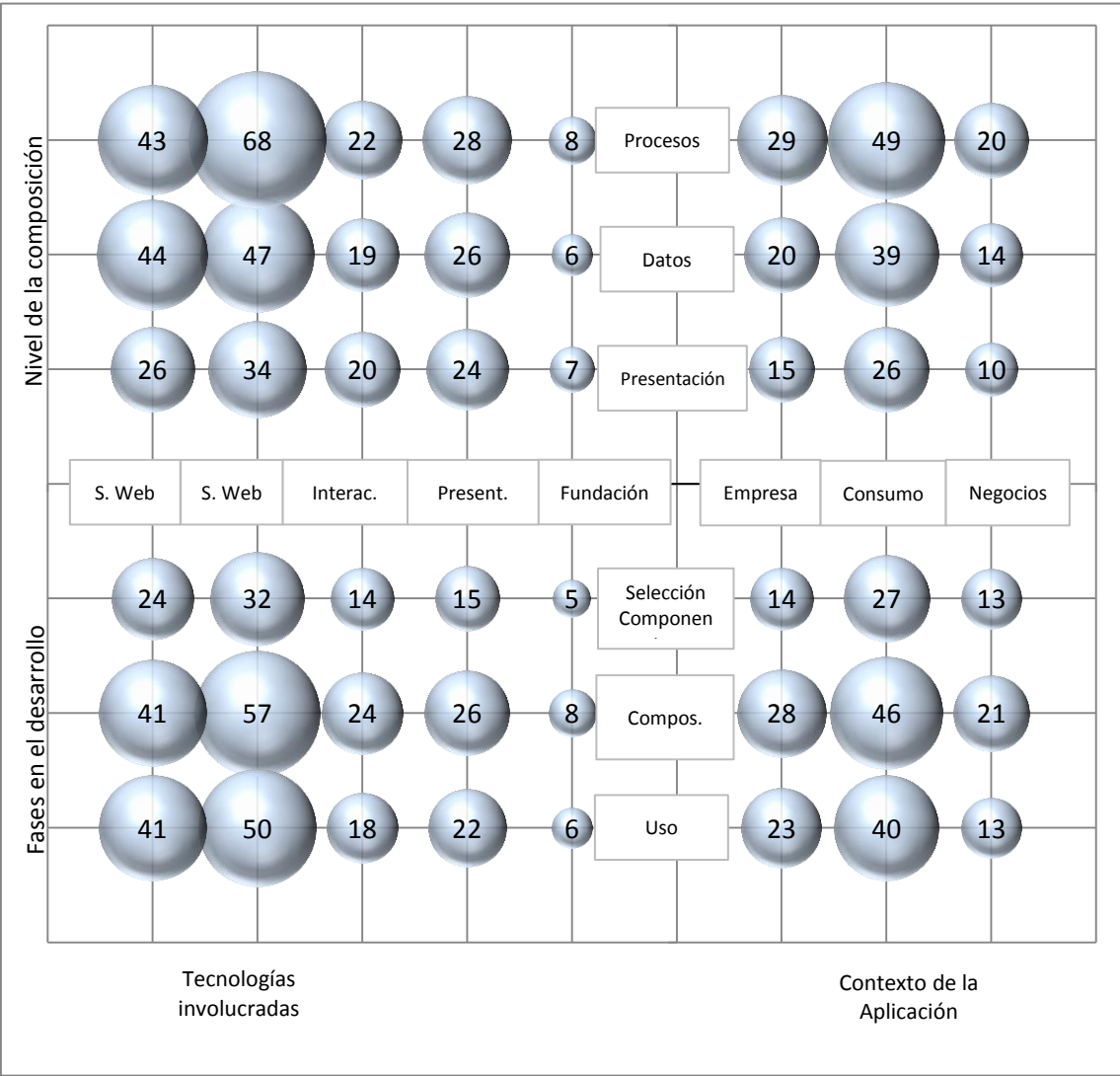


Figura 3-7 Relaciones entre criterios de extracción. Mapeo Sistemático Composicionalidad

Dificultades en la Validación:

Las principales dificultades de este estudio son: el sesgo en el ámbito de nuestras preguntas de investigación, publicación y selección, la imprecisión en la extracción de datos y errores en la clasificación.

Los sesgos de publicación se refieren al problema de que los resultados positivos son más deseables de publicación que los negativos [21]. Nosotros estamos conscientes de que esta es

una limitación inherente a nuestras fuentes bibliográficas. Con respecto al sesgo en la selección de publicaciones, nosotros escogimos las fuentes donde los estudios sobre el desarrollo de Mashups son normalmente publicados y comparamos los estudios obtenidos contra una pequeña muestra la cual fue previamente identificada como estudios relevantes. Sin embargo, nosotros no consideramos otras fuentes bibliográficas tales como Springer Link o ScienceDirect lo que puede haber afectado a la completitud de nuestro mapeo sistemático. Además, nuestra búsqueda bibliográfica fue conducida al final del año 2012, algunos estudios no estuvieron aún indexados en ese momento y pudieron no haber sido considerados.

Finalmente, nosotros intentamos aliviar la imprecisión en la extracción de los datos y la falta de una adecuada clasificación por medio de tres revisores y la resolución de discrepancias por consenso.

Capítulo 4. Modelo de Usabilidad para Mashups

Los Mashups han ganado popularidad como un resultado del balance entre el costo de desarrollo, la facilidad de integración de componentes de terceros y la calidad de sus soluciones. Existen algunas características específicas para los Mashups que los hacen diferentes de las aplicaciones Web tradicionales, como por ejemplo su esencia composicional. Los Mashups raramente tienen el concepto de navegación por cuanto son usualmente presentados sobre una página Web individual. Es más, ellos son creados para situaciones específicas y a menudo empleados por períodos cortos de tiempo. Estas características específicas han hecho necesario el contar con un modelo de usabilidad específico para este tipo de aplicaciones.

En esta tesina, se propone un Modelo de Usabilidad de Mashups que está alineado con el estándar ISO / IEC 2501 (SQuaRE) y extiende el Modelo de Usabilidad Web para aplicaciones tradicionales presentado por [19] para Mashups. Este modelo de usabilidad es usado para evaluar y mejorar la usabilidad de los Mashups considerando sus características particulares.

Algunos modelos de calidad para evaluaciones de usabilidad Web han sido propuestos en los últimos años. Estos modelos han sido definidos basándose en estándares existentes tales como ISO / IEC 9241-1 y también ISO / IEC 9126-1.

Los modelos de calidad para la Web son derivados desde estándares existentes, entre estos modelos tenemos los de Olsina y Rossi [50], Calero et al. [10], Seffahet et al. [60] y Moraga et al. [48].

Olsina y Rosi [50], proponen un método de evaluación de calidad para la Web (WebOEM) con el cual se definen las características y atributos de calidad basados en la ISO / IEC 9126, tales como la usabilidad, funcionalidad, confiabilidad y efectividad; las necesidades de los usuarios Web son además incorporados. Calero [10] presenta el modelo de calidad Web (WQM), en el cual tres dimensiones son distinguidas: Características Web (contenido, presentación y navegación); características de calidad basadas en el ISO / IEC 9126 (funcionalidad, confiabilidad, usabilidad, eficiencia, portabilidad y mantenibilidad); y los procesos de ciclo de vida desde el ISO / IEC 12207 (desarrollo, operación y mantenimiento). Procesos organizacionales, tales como manejo de proyectos y programas de reutilización. Con respecto a la nueva generación de aplicaciones Web, Olsina [49], actualizó la primera versión del 2Q2U (Calidad, Calidad en uso, usabilidad actual y experiencia de usuario) modelando un marco de trabajo, destinado a aplicaciones Web tradicionales, considerando características específicas para aplicaciones Web modernas (Web 2.0) como lo son la integración, maximización de beneficios, comunicabilidad y sentido de comunidad.

Cappiello et al. [13], presenta un enfoque que trata de la información de la calidad en los Mashups. El muestra que los usuarios del Mashup no están interesados en cómo el Mashup fue construido, sino simplemente quieren que la aplicación del Mashup se ejecute como se espera, sin datos incorrectamente alineados, faltantes o problemas de información similares. En otras palabras, el usuario está interesado en percibir la calidad externa. Estos autores definen la calidad de los Mashups en términos de calidad de componentes, calidad en la composición y calidad en la información. Los autores se mantienen en que la categoría de la

calidad de la composición “es muy peculiar para los Mashups por cuanto estos se centran en componentes que interactúan entre ellos y miden la utilidad de esa composición).

En Capiello et al. [11], se propone un modelo de calidad que enfatiza en la naturaleza basada en componentes de los Mashups y se centra particularmente en la calidad de la composición. Sin embargo, para mejorar nuestro conocimiento, no existen guías de cómo evaluar la usabilidad de los Mashups considerando sus características intrínsecas. Además no son modelos de calidad basados en estándares que faciliten las evaluaciones de calidad de esta clase de aplicaciones Web.

Aquí se provee un modelo de calidad de las principales sub-características, atributos y ciertas métricas del modelo de Usabilidad. Este modelo es una adaptación y extensión del modelo de usabilidad Web provisto por Fernández [19], el cual descompone el concepto de usabilidad de las aplicaciones Web en seis sub-características propuestas en el estándar ISO / IEC 25000 SQuaRE, el cual está compuesto por otras sub-características y atributos que consideran el criterio ergonómico propuesto en [5] y las guías de usabilidad para el desarrollo Web [39]. Nosotros, hemos adaptado la definición de estas sub-características y atributos y además hemos agregado nuevas, que cubran la naturaleza de los Web Mashups.

SQuaRE propone que, la usabilidad de un producto software, puede ser descompuesta en las siguientes características: **Facilidad de entendimiento** (Appropriateness recognisability), **Facilidad de aprendizaje** (Learnability), **Facilidad de uso** (Ease of use), **Facilidad de ayuda** (Helpfulness), **Accesibilidad técnica** (Technical accessibility), **Grado de atracción** (Attractiveness), **y Adherencia a normas o convenciones** (Compliance). Sin embargo, estas características son demasiado abstractas y necesitan ser descompuestas en atributos más fácilmente medibles.

En [19], se propone un modelo de calidad basado en SQuaRE, específico para aplicaciones Web, este modelo ha sido tomado como base para este trabajo y ha sido extendido para abordar el carácter composicional de los Mashups.

Tanto en [31][19], se muestran las cinco primeras sub-características (facilidad de entendimiento, facilidad de aprendizaje, facilidad de ayuda, facilidad de uso y accesibilidad técnica) las mismas que están relacionadas con el rendimiento del usuario hacia la aplicación y pueden ser cuantificadas de una manera más objetiva.

4.1. Facilidad de Entendimiento (Appropriateness recognisability)

La facilidad de entendimiento se refiere a las características del Mashup que hacen más fácil el entendimiento del usuario con respecto a la aplicación.

En el modelo base [19], ésta característica se ha descompuesto diferenciando los atributos que permiten la **legibilidad visual** de textos e imágenes de aquellos que facilitan la lectura de la información en lo que se refiere a la densidad y agrupación de la información. La legibilidad en cuanto a lo que se puede leer envuelve información que agrupa cohesión, densidad de información y soporte a la paginación. Además se incluyen características tales como la **familiaridad**, la facilidad con la cual un usuario reconoce los componentes de la interface y ve su interacción con estos de manera natural. La **reducción de esfuerzo** del usuario, que se relaciona con la reducción del esfuerzo cognitivo del usuario, la guía de usuario, la cual está

relacionada a la disponibilidad de mensajes y la retroalimentación informativa en respuesta a las acciones del usuario y la navegabilidad, la cual está relacionada a cómo el cometido está accedido por el usuario.

(1.1) Legibilidad Visual

Además de los atributos propios de las aplicaciones Web en general, se han agregado atributos específicos de los Mashups, en los cuales su carácter composicional hace necesaria una legibilidad visual que permita encontrar y distinguir fácilmente la información en los componentes provistos por terceras partes, para ello se han agregado:

- **(1.1.4) Disposición de Componentes (DC)**, está relacionado al modo en el cual los componentes son organizados en la página Web para su identificación visual. Es importante mantener un balance en la distribución de componentes sobre la página Web, para de esta manera evitar una sobrecarga de información en regiones o áreas y para evitar el solapamiento de los componentes. Este atributo puede ser medido considerando la relación de los componentes visibles (número de componentes totalmente visibles) entre el número total de componentes, el cual se refiere a los componentes que son completamente visibles (sin solapamiento) y la relación de la medición de regiones útiles del Mashup (número de regiones usadas en la pantalla para el número total de regiones en la pantalla).

$$DC = \frac{\text{Número de componentes totalmente visibles}}{\text{Número total de componentes a mostrarse}} = \text{valor}$$

De acuerdo al grado de impacto de este atributo en la usabilidad del Mashup se establecen los siguientes umbrales:

Problema mayor de usabilidad:	$0 \leq \text{valor} < 0.35$
Problema medio de usabilidad:	$0.35 \leq \text{valor} < 0.75$
Problema menor de usabilidad:	$0.75 \leq \text{valor} < 1$
Sin ningún problema de usabilidad:	1

La interpretación de la métrica es $0 \leq \text{valor} \leq 1$ en donde el valor más cercano a 1 es el mejor.

- **(1.1.5) Tamaño del Componente (TC)**, este atributo está relacionado al tamaño de los componentes para su visualización y uso. Los componentes podrían ser útiles y usables a los usuarios solamente si ellos pueden ser fácilmente manipulados y si pueden mostrar toda la información necesaria sin ningún esfuerzo por parte del usuario. La forma en la cual las métricas para este atributo son definidas y calculadas dependen del tipo de componente y del tipo de información que los componentes manejan (por ejemplo un mapa o mensajes). Una métrica para este atributo es la relación de componentes con tamaño apropiado (número de componentes con tamaño apropiado para visualizar información relevante y controles entre el número total de componentes).

$$TC = \frac{\text{Número de componentes con tamaño apropiado}}{\text{Número total de componentes}} = \text{valor}$$

De acuerdo al grado de impacto de este atributo en la usabilidad del Mashup se establecen los siguientes umbrales:

Problema mayor de usabilidad:	$0 \leq \text{valor} < 0.35$
Problema medio de usabilidad:	$0.35 \leq \text{valor} < 0.75$
Problema menor de usabilidad:	$0.75 \leq \text{valor} < 1$
Sin ningún problema de usabilidad:	1

La interpretación de la métrica es $0 \leq \text{valor} \leq 1$ en donde el valor más cercano a 1 es el mejor.

(1.2) Legibilidad

Dentro de la sub-característica legibilidad se han agregado dos atributos nuevos relacionados directamente con los Mashups:

- **(1.2.1) Coherencia en la Agrupación de los Componentes (CAC)**, en la cual se observa si los componentes están agrupados coherentemente de acuerdo al propósito de estos. La métrica asociada es la relación del número total de componentes coherentes entre el número total de componentes.

$$CAC = \frac{\text{Número de componentes coherente}}{\text{Número total de componentes}} = \text{valor}$$

En el caso de este atributo el valor que toma es de 1 ya que al aportar todos los componentes un propósito que contribuya al objetivo del Mashup su relación quedaría en 1. No presentándose ningún problema de usabilidad.

De acuerdo al grado de impacto de este atributo en la usabilidad del Mashup se establecen los siguientes umbrales:

Problema mayor de usabilidad:	$0 \leq \text{valor} < 0.35$
Problema medio de usabilidad:	$0.35 \leq \text{valor} < 0.75$
Problema menor de usabilidad:	$0.75 \leq \text{valor} < 1$
Sin ningún problema de usabilidad:	1

La interpretación de la métrica es $0 \leq \text{valor} \leq 1$ en donde el valor más cercano a 1 es el mejor.

(1.2.2) Densidad de la Información (DI), que tiene que ver con la cantidad de información mostrada al mismo tiempo, lo que puede hacer que la aplicación se muestre sobrecargada.

Con respecto a la densidad de la información, la misma que tiene que ver con la cantidad de información mostrada al mismo tiempo, establecimos una escala de Likert en donde realizamos la siguiente pregunta:

La información se muestra en una forma adecuada y sin sobrecargar la página.

1. Totalmente de acuerdo.
2. Ni en acuerdo, ni en desacuerdo.
3. Totalmente en desacuerdo.

En este caso si en la evaluación se muestra la opción 3, se presenta un problema de usabilidad; en el caso de la opción 2, se presenta un problema menor de usabilidad; y por último en primer caso no existe problema ninguno de usabilidad

(1.3) Familiaridad

Dentro de la familiaridad a más de los atributos del modelo base tenemos

- **(1.3.4) Popularidad de los Componentes (PC)**, en la cual se comprueba si los componentes son conocidos, aceptados y comúnmente usados en la comunidad y está medida por la relación existente entre el número de componentes bien conocidos y el número total de componentes.

$$PC = \frac{\text{Número de componentes bien conocidos}}{\text{Número total de componentes}} = \text{valor}$$

Para este atributo, debemos tener en cuenta que no necesariamente el hecho de que el componente no sea muy conocido pueda afectar a la usabilidad, ya que un componente no tan conocido puede ser muy fácil e intuitivo de usar, sin embargo el usar componentes bien conocidos puede aportar al éxito del Mashup, ya que son componentes aceptados por la comunidad de usuarios, que no necesitan un aprendizaje inicial y que ayudan a darle al Mashup una aceptación más rápida. Por esta razón los rangos establecidos son diferentes a los que hemos venido utilizando en atributos anteriores, teniendo:

Problema medio de usabilidad: $0 \leq \text{valor} < 0.5$

Problema menor de usabilidad: $0.5 \leq \text{valor} < 1$

Sin ningún problema de usabilidad: 1

La interpretación de la métrica es $0 \leq \text{valor} \leq 1$ en donde el valor más cercano a 1 es el mejor.

- **(1.3.5) Presentación de Controles en Componentes (PCC)**, en el cual se toma en cuenta si los controles en los componentes son usuales, midiendo este atributo a través de la relación existente entre los componentes con controles usuales y el número total de componentes que presentan controles.

$$PCC = \frac{\text{Controles usuales}}{\text{Nro.Total de Componentes que presentan controles}} = \text{valor}$$

Este caso, es muy similar al atributo anterior, el hecho de que los controles sean usuales aporta un valor agregado al componente, sin embargo eso no significa que si utilizáramos un componente con controles distintos necesariamente representara un problema de usabilidad.

Los umbrales elegidos en este caso son:

Problema medio de usabilidad: $0 \leq valor < 0.7$

Problema menor de usabilidad: $0.7 \leq valor < 1$

Sin ningún problema de usabilidad: 1

La interpretación de la métrica es $0 \leq valor \leq 1$ en donde el valor más cercano a 1 es el mejor.

(1.4) Reducción de Esfuerzo

Con respecto a la reducción de esfuerzo, los atributos que se han agregado al modelo de usabilidad relacionados a los Mashups son:

- **(1.4.4) Actualización de Información entre Componentes (AIC)**, en este atributo se evalúa si los componentes interconectados son actualizados automáticamente, esto es, en el caso de que se refresque por una búsqueda o acción algún componente, si los otros que están vinculados y deben ser actualizados con la nueva información, se refrescan automáticamente sin necesidad de la intervención del usuario. La métrica para medir este atributo es la proporción de componentes auto-sincronizados, entre el número total de componentes que comparten algunos datos con otros componentes y que deben ser sincronizados.

$$AIC = \frac{\text{Componentes auto_sincronizados}}{\text{Nro. total componentes que deben autosincronizarse}} = valor$$

De acuerdo al grado de impacto de este atributo en la usabilidad del Mashup se establecen los siguientes umbrales:

Problema mayor de usabilidad: $0 \leq valor < 0.35$

Problema medio de usabilidad: $0.35 \leq valor < 0.75$

Problema menor de usabilidad: $0.75 \leq valor < 1$

Sin ningún problema de usabilidad: 1

La interpretación de la métrica es $0 \leq valor \leq 1$ en donde el valor más cercano a 1 es el mejor.

Se ha omitido la completitud del mapa del sitio, debido a que los Mashups generalmente se presentan en una sola página, por lo que no hace falta un mapa del sitio.

A continuación se muestra una tabla que hace referencia a la facilidad de entendimiento del Modelo de calidad en el que se han agregado sub-características y atributos propios de los Mashups y que en secciones posteriores se indicará como ayudan a la evaluación de este tipo de productos. Se ha resaltado todos aquellos atributos que se han agregado y que permitirán evaluar específicamente Mashups.

Sub-características	Atributos	Explicación/Heurística
1.1. Legibilidad visual	1.1.1. Tamaño de fuente /adecuación visual	¿La fuente (color, tipo, tamaño) al contexto es adecuada?

Sub-características	Atributos	Explicación/Heurística
	1.1.2. Adecuación de la visualización del texto.	¿El color del contraste entre el texto y el fondo es adecuado?
	1.1.3. Disposición del texto	¿Los textos relevantes son fáciles de encontrar sin necesidad de acciones adicionales? (no existen barras horizontales)
	1.1.4. Disposición de componentes	¿Los componentes son fáciles de encontrar y reconocer en la página Web?
	1.1.5. Tamaño del componente	¿Los tamaños de los componentes son apropiados para ver y usar cómodamente?
1.2. Legibilidad	1.2.1. Coherencia en la agrupación de componentes	¿Los componentes están agrupados coherentemente de acuerdo a su propósito?
	1.2.2. Densidad de la información	Relacionado con 1.4.3
1.3. Familiaridad	1.3.1 Consistencia en el formato de los datos.	¿Los conceptos siempre usan la misma representación o notación?
	1.3.2. Metáfora	¿Los íconos/acciones representan conceptos/actividades relacionadas a la vida real?
	1.3.3. Internacionalización	¿Usa de elementos y formas que siguen estándares?
	1.3.4. Popularidad de componentes.	¿Los componentes son conocidos, aceptados y comúnmente usados en la comunidad?
	1.3.5. Presentación de controles en componentes.	¿Son usuales los controles en los componentes?
1.4. Reducción de esfuerzo	1.4.1. Acciones mínimas	¿Se proveen mecanismos para minimizar las acciones de usuarios? (valores por defecto, resultados recientes, etc.)
	1.4.2. Auto-descripción	¿Los campos de entrada son entendibles?
	1.4.3. Complejidad de la información.	¿Existe mucha información mostrada al mismo tiempo?
	1.4.4. Actualización de información entre componentes	¿Los componentes interconectados son actualizados automáticamente?
1.5. Orientación al usuario	1.5.1. Disponibilidad de mensajes	¿Existen mensajes que guíen la interacción del usuario (mensajes de error, aviso y advertencia)?
	1.5.2. Retroalimentación inmediata	¿Se proveen acciones de retroalimentación para acciones inmediatas? (campos resaltados, mensajes con tips, animación de botones, cursores de carga, etc.)
	1.5.3. Contexto de usuario explícito	¿Existe capacidad de proveer el contexto en el cuál se está localizado con la aplicación Web?
1.6. Navegabilidad	1.6.1. Soporte de búsqueda interna	¿Se provee soporte de búsqueda interno?
	1.6.2. Clickabilidad	¿Los vínculos son fácilmente detectados como tales?
	1.6.3. Interconectividad	¿Existe un buen grado de interconexión entre los contenidos y/o acciones de la aplicación Web?
	1.6.4. Alcanzabilidad	¿Existe facilidad de acceder a los contenidos y/o acciones de la aplicación Web?

Tabla 4-1 *Facilidad de entendimiento.*

4.2. Facilidad de Aprendizaje (Learnability)

La facilidad de aprendizaje mide cuan fácil es que un usuario nuevo se convierta en experto utilizando un sistema. Esta característica hace referencia a todos los atributos presentes en la aplicación Web, en este caso en los Mashups, que hacen posible que el usuario aprenda a

usarlo y qué tan fácil es este proceso. En el modelo de calidad para Mashups propuesto en este trabajo y dado que este es tomado como base de [19], en la cual se ha descompuesto en las sub-características:

La **Predictibilidad** se refiere a la facilidad con la cual el usuario puede predecir las acciones en base a la interface con la que cuenta. La **potencialidad**, se refiere a la facilidad con la cual un usuario descubre qué acciones se pueden realizar en el siguiente paso de su interacción con el sistema. La **retroalimentación informativa**, se refiere a la información que se obtiene a cambio de una acción como resultado de la misma. Como menciona [19], estas características son muy importantes dentro del ámbito de las aplicaciones Web ya que el usuario no cuenta con mucho tiempo para aprender el funcionamiento del sistema, destacándose adicionalmente que en lo que a Mashups se refiere, debido a su carácter situacional y de corto tiempo de vida, no merecería la pena un difícil aprendizaje del sistema, puesto que no aportaría mucho en la solución del problema puntual, el tener que usar demasiado tiempo en el aprendizaje de la herramienta.

(2.1) Predictibilidad

En cuanto a la predictibilidad, se ha agregado:

- **(2.1.4) Propósito Fácilmente Distinguible de los Componentes (PFD)**, este atributo se mide por medio de la relación de componentes distinguibles entre el número total de componentes, y trata sobre la facilidad de que el usuario entienda cual es el propósito del componente dentro del Mashup de una manera sencilla, la métrica utilizada y mostrada de forma más concisa es:

$$PFD = \frac{\text{Nro. componentes fácilmente distinguibles}}{\text{Nro. total de componentes}} = \text{valor}$$

La valoración de la usabilidad la haremos utilizando los siguientes umbrales:

Problema mayor de usabilidad:	$0 \leq \text{valor} < 0.35$
Problema medio de usabilidad:	$0.35 \leq \text{valor} < 0.75$
Problema menor de usabilidad:	$0.75 \leq \text{valor} < 1$
Sin ningún problema de usabilidad:	1

La interpretación de la métrica es $0 \leq \text{valor} \leq 1$ en donde el valor más cercano a 1 es el mejor.

(2.2) Potencialidad

Dentro de la potencialidad y atendiendo las características particulares de los Mashups, se ha agregado:

- **(2.2.3) Distinción de Posibles Acciones de los Componentes (DPA)**, el mismo que permite evaluar si los componentes muestran sus acciones claramente dentro del Mashup. La métrica asociada relaciona las acciones de los componentes cuyas acciones son fácilmente identificables entre el número total de componentes. Cuando nos

referimos a las acciones de los componentes trata sobre la facilidad de la identificación de todas las acciones del componente dentro del Mashup.

$$DPA = \frac{\text{Acciones de componentes fácilmente identificadas}}{\text{Nro. total de componentes}} = \text{valor}$$

La valoración de la usabilidad se realiza utilizando los siguientes umbrales:

- Problema mayor de usabilidad: $0 \leq \text{valor} < 0.35$
- Problema medio de usabilidad: $0.35 \leq \text{valor} < 0.75$
- Problema menor de usabilidad: $0.75 \leq \text{valor} < 1$
- Sin ningún problema de usabilidad: 1

La interpretación de la métrica es $0 \leq \text{valor} \leq 1$ en donde el valor más cercano a 1 es el mejor.

La tabla que se muestra a continuación, muestra de forma detallada la descomposición de las sub-características mencionadas en atributos que puedan ser medidos más fácilmente, destacándose aquellos que tienen directamente que ver con los Mashups.

Atributos	Atributos	Explicación/Heurística
2.1. Predictibilidad	2.1.1. Nombres de enlaces significativos	¿Es factible predecir la acción que se va a realizar atendiendo al nombre de los enlaces?
	2.1.2. Etiquetas significativas	¿Es fácil predecir qué concepto representa una etiqueta asociada a cualquier elemento de la interfaz?
	2.1.3. Controles significativos	¿Es fácil predecir qué acción realizará un control determinado?
	2.1.4. Propósito fácilmente distinguible de los componentes	¿Los componentes contribuyen al propósito del Mashup?
2.2. Potencialidad	2.2.1. Determinación de posibles acciones	¿Son fáciles de encontrar todas las acciones posibles?
	2.2.2. Determinación de acciones prometedoras	¿Las acciones más relevantes son fáciles de encontrar?
	2.2.3. Distinción de posibles acciones de los componentes.	¿Los componentes muestran sus acciones claramente?
2.3. Retroalimentación informativa	2.3.1. Progreso explícito de las transacciones	¿Es capaz la aplicación Web de proporcionar a los usuarios el estado de las transacciones que se realizan? (tareas completadas con éxito, estado en una transacción, etc.).
	2.3.2. Contexto explícito del usuario	¿Es capaz la aplicación de proporcionar a los usuarios el contexto en el que se encuentran dentro de la aplicación Web? (Sesión iniciada, nivel de privacidad de la información, etc.).
	2.3.3. Visibilidad de resultados	¿Se muestran los resultados de las acciones de los usuarios?
2.4. Ayuda	2.4.1. Ayuda en línea completa	¿La ayuda en línea es completa?
	2.4.2. Documentación multi-usuario	¿Existe documentación multi-usuario?

Tabla 4-2 *Facilidad de aprendizaje.*

4.3. Operatividad

La Operatividad hace referencia a los atributos de la aplicación Web que permiten la adecuada operación y control de la misma. En el modelo de usabilidad propuesto en [19] que tenemos como base, se ha descompuesto en otras características que tienen que ver más directamente con las aplicaciones Web, y en el presente trabajo se han añadido aquellas que tienen específicamente relación con los Mashups, redefiniéndose algunas para que se enmarquen más en el dominio aquí tratado. La definición viene desde la “controlabilidad, tolerancia a fallos y conformidad con las expectativas del usuario”, conceptos que se han definido en el ISO / IEC 9241-10 (1996).

Dentro del modelo de usabilidad base, estas sub-características fueron divididas en otras sub-características relacionadas a otros aspectos técnicos de aplicaciones Web tales como el manejo de los datos de entrada, controlabilidad, capacidad de adaptación y consistencia, aquí hemos modificado la compatibilidad presentada en el modelo general de aplicaciones Web anteriormente citado por la sub-característica ejecuciones portables, que engloba también la portabilidad que puede necesitar un Mashup.

(3.2) Controlabilidad

Dentro de la controlabilidad y relacionada a los Mashups encontramos.

- **(3.2.8) Facilidad para Alcanzar los Componentes (FAC)**, la misma que nos permite evaluar si el usuario puede alcanzar todos los componentes en el Mashup, siendo su métrica la relación existente entre los componentes alcanzables (número de componentes alcanzables) entre el número total de componentes.

$$FAC = \frac{\text{Componentes alcanzables}}{\text{Nro. total componentes}} = \text{valor}$$

En el caso de este atributo tenemos los siguientes umbrales:

Problema mayor de usabilidad:	$0 \leq \text{valor} < 0.35$
Problema medio de usabilidad:	$0.35 \leq \text{valor} < 0.75$
Problema menor de usabilidad:	$0.75 \leq \text{valor} < 1$
Sin ningún problema de usabilidad:	1

La interpretación de la métrica es $0 \leq \text{valor} \leq 1$ en donde el valor más cercano a 1 es el mejor.

- **(3.2.9) Carga de los Componentes (CC)**, en este atributo se habla de la dificultad en la carga de los mismos, ya sea por su tamaño o por alguna restricción que haga difícil esta acción, se mide la carga de los componentes por medio del número de componentes que presentan dificultad en su carga entre el número total de componentes.

$$CC = \frac{\text{Nro. componentes sin dificultad de carga}}{\text{Nro. total componentes}} = \text{valor}$$

Al ser los umbrales los siguientes:

Problema mayor de usabilidad: $0 \leq valor < 0.35$

Problema medio de usabilidad: $0.35 \leq valor < 0.75$

Problema menor de usabilidad: $0.75 \leq valor < 1$

Sin ningún problema de usabilidad: 1

La interpretación de la métrica es $0 \leq valor \leq 1$ en donde el valor más cercano a 1 es el mejor.

(3.4) Consistencia

Dentro de la consistencia encontramos.

- **(3.4.6) Comportamiento Correcto de los Componentes (CCC)**, en este atributo se evalúa si los componentes se comportan correctamente de acuerdo a sus entradas de datos/eventos, esta evaluación se realiza mediante la relación de correcto comportamiento de componentes (número de componentes con el comportamiento apropiado) entre el número total de componentes.

$$CCC = \frac{\text{Nro. componentes comportamiento apropiado}}{\text{Nro. total componentes}} = valor$$

Los umbrales establecidos para esta métrica son:

Problema mayor de usabilidad: $0 \leq valor \leq 0.5$

Problema medio de usabilidad: $0.5 < valor \leq 0.9$

Problema menor de usabilidad: $0.9 < valor < 1$

No existe un problema de usabilidad 1

La interpretación de la métrica es $0 \leq valor \leq 1$ en donde el valor más cercano a 1 es el mejor.

(3.5) Ejecuciones Portables

Hemos agregado la sub-característica ejecuciones portables, la misma que está conformada por los atributos:

- **(3.5.1) Independencia de Dispositivos (ID)**, en la cual se comprueba si la operatividad de los Mashups es la misma a través de diferentes plataformas, para ello se utiliza la métrica de la relación existente entre las plataformas en las que el Mashup conserva la misma operatividad (número de plataformas con la misma operatividad del Mashup) entre el número total de plataformas para las que se espera que el Mashup funcione de la misma manera.

$$ID = \frac{\text{Nro. plataformas que conservan operatividad}}{\text{Nro. total de plataformas en que se espera igual operatividad}} = valor$$

Los umbrales establecidos para esta métrica son:

Problema mayor de usabilidad: $0 \leq valor < 0.35$

Problema medio de usabilidad:	$0.35 \leq valor < 0.75$
Problema menor de usabilidad:	$0.75 \leq valor < 1$
Sin ningún problema de usabilidad:	1

La interpretación de la métrica es $0 \leq valor \leq 1$ en donde el valor más cercano a 1 es el mejor.

- **(3.5.2) Auto-ajuste de los Componentes en los Dispositivos (ACD)**, se deberá comprobar si los componentes cambian de tamaño de acuerdo a las diferentes plataformas, dispositivos o tamaño de pantalla, pudiendo ser este atributo evaluado por medio de la relación existente entre el número de resoluciones de pantalla compatibles en las cuales los componentes se auto-ajustan y el número total de las pantallas más populares en las que se desea tener portabilidad.

$$ACD = \frac{\text{Resoluciones en las que se auto - ajustan los componentes}}{\text{Nro de resoluciones en las que se desea portabilidad}} = valor$$

Los umbrales establecidos para esta métrica son:

Problema mayor de usabilidad:	$0 \leq valor < 0.35$
Problema medio de usabilidad:	$0.35 \leq valor < 0.75$
Problema menor de usabilidad:	$0.75 \leq valor < 1$
Sin ningún problema de usabilidad:	1

La interpretación de la métrica es $0 \leq valor \leq 1$ en donde el valor más cercano a 1 es el mejor.

- **(3.5.3) Facilidad para Instalar Software Adicional (FIS)**, toma en cuenta cuándo los componentes necesitan instalar complementos para poder ser usados, qué tan difícil es esta acción, se puede evaluar por medio del número de componentes que necesitan software adicional y que no puede ser instalado fácilmente. Para realizar esta medición se podrá usar la siguiente fórmula.

$$ACD = \frac{\text{Componentes con dificultad de ejecución por software adicional}}{\text{Número total de componentes que requieren software adicional}} = valor$$

Los umbrales establecidos para esta métrica son:

Problema mayor de usabilidad:	$0 \leq valor < 0.35$
Problema medio de usabilidad:	$0.35 \leq valor < 0.75$
Problema menor de usabilidad:	$0.75 \leq valor < 1$
Sin ningún problema de usabilidad:	1

La interpretación de la métrica es $0 \leq valor \leq 1$ en donde el valor más cercano a 1 es el mejor.

- **(3.5.4) Compatibilidad con la Velocidad de Conexión (CVC)**, en el cuál se evalúa si el Mashup tiene la capacidad de adaptarse en su ejecución a las más comunes velocidades de conexión, por ejemplo Wi-fi, 3G, etc.

$$ACD = \frac{\text{Componentes con ejecución lenta por conexion}}{\text{Número total de componentes}} = \text{valor}$$

Los umbrales establecidos para esta métrica son:

- Problema mayor de usabilidad: $0 \leq \text{valor} < 0.35$
- Problema medio de usabilidad: $0.35 \leq \text{valor} < 0.75$
- Problema menor de usabilidad: $0.75 \leq \text{valor} < 1$
- Sin ningún problema de usabilidad: 1

La interpretación de la métrica es $0 \leq \text{valor} \leq 1$ en donde el valor más cercano a 1 es el mejor.

La tabla a continuación muestra más en detalle todos los atributos en los cuales esta sub-característica ha sido dividida, resaltando aquellos más específicos para los Mashups.

Atributos	Atributos	Explicación/Heurística
3.1. Manejo de los datos de entrada	3.1.1. Validación de los datos de entrada	¿Existen mecanismos para prevenir entradas de datos no válidas? (fechas, números, valores restringidos)
	3.1.2. Privacidad de los datos	¿Existen mecanismos para prevenir las entradas de datos privadas? (campos de contraseñas enmascarados)
3.2. Controlabilidad	3.2.1. Aplazamiento de la edición	¿Se puede editar el valor de un campo de entrada una vez que se ha enviado la información al servidor?
	3.2.2. Soporte a la cancelación	¿Está permitido cancelar las operaciones de usuario antes de que ellas sean ejecutadas?
	3.2.3. Soporte de interrupción	¿Está permitido interrumpir las operaciones de usuario mientras ellas son ejecutadas?
	3.2.4. Soporte para deshacer	¿Está permitido deshacer las operaciones de usuario después de que se ejecuten?
	3.2.5 Soporte para rehacer	¿Está permitido rehacer acciones cuando sea necesario?
	3.2.6 Soporte para filtrado	¿Se permite filtrar la información mostrada?
	3.2.7 Soporte para formato de impresión	¿Está permitido presentar la información para impresión en un modo amigable?
	3.2.8 Facilidad para alcanzar los componentes	¿Puede el usuario alcanzar todos los componentes en el Mashup?
	3.2.9. Carga de los componentes	¿La carga de los componentes es difícil debido a su tamaño? ¿Existe alguna restricción que haga la carga de componentes difícil?
3.3. Capacidad de Adaptación	3.3.1 Adaptabilidad	¿La aplicación Web tiene la habilidad de adaptarse a los usuarios?

Atributos	Atributos	Explicación/Heurística
	3.3.2 Facilidad de adaptación a usuarios	¿La aplicación Web puede reunir las necesidades de diferentes usuarios?
3.4. Consistencia	3.4.1 Comportamiento constante de vínculos / controles	¿Cada vínculo /control tiene el mismo comportamiento a través del Mashup?
	3.4.2 Permanencia de controles / vínculos	¿Los vínculos/controles aparecen si se pueden realizar las acciones asociadas?
	3.4.3 Consistencia en el orden de los vínculos/controles	¿Los vínculos/controles aparecen si sus acciones asociadas pueden ser ejecutadas?
	3.4.4 Consistencia en el orden de los controles	¿Cada control aparece en el mismo lugar a través de todo el Mashup?
	3.4.5 Consistencia de las etiquetas / encabezamientos	Los encabezamientos /etiquetas corresponden a las acciones las cuales son ejecutadas para acceder a las mismas.
	3.4.6. Comportamiento correcto de los componentes.	¿Los componentes se comportan correctamente de acuerdo a sus entradas datos / eventos?
3.5. Ejecuciones portables	3.5.1 Independencia de dispositivos	¿La operatividad de los Mashups es la misma a través de diferentes plataformas?
	3.5.2. Auto-ajuste de los componentes en los dispositivos	¿Los componentes cambian de tamaño de acuerdo a las diferentes plataformas, dispositivos o tamaño de pantalla?
	3.5.3. Facilidad para instalar software adicional necesario	¿Cuándo los componentes necesitan complementos, cuán difícil es instalarlos?
	3.5.4. Compatibilidad con velocidades de conexión.	¿El Mashup tiene la capacidad de adaptarse en su ejecución a las más comunes velocidades de conexión? (e.g. Wifi, 3G).

Tabla 4-3 Atributos de la sub-característica "Operatividad"

4.4. Protección de Errores al Usuario

En cuanto a la protección de errores al usuario, se habla de la prevención y recuperación de errores, haciendo alusión a si el Mashup puede prevenir y/o recuperarse de los errores de usuario que se puedan dar a lo largo de su ejecución. Para ello, contamos con un atributo relacionado a esta sub-característica.

- **(4.3) Reemplazo de componentes (RC)**, evalúa si existe un reemplazo para componentes que presenten algún inconveniente en su ejecución hace alusión a que si el Mashup tiene algún mecanismo semántico o de cualquier otra índole que entre en juego al momento de que un componente esté ausente, fuera de servicio o presente algún error de carga. O en caso contrario se presenta algún aviso al respecto, de tal manera que el usuario sepa claramente de que existe allí un componente que pronto entrará en funcionamiento. La métrica que se utiliza para comprobar si este atributo se cumple o no es la relación entre el número de componentes que son reemplazados y el número total de componentes en el Mashup.

$$RC = \frac{\text{Componentes que pueden ser reemplazados}}{\text{Nro total de componentes}} = \text{valor}$$

Los umbrales establecidos para esta métrica son:

Problema mayor de usabilidad:	$0 \leq \text{valor} < 0.35$
Problema medio de usabilidad:	$0.35 \leq \text{valor} < 0.75$
Problema menor de usabilidad:	$0.75 \leq \text{valor} < 1$
Sin ningún problema de usabilidad:	1

La interpretación de la métrica es $0 \leq \text{valor} \leq 1$ en donde el valor más cercano a 1 es el mejor.

Se presenta una tabla con la sub-característica de protección de errores al usuario:

Atributos	Explicación/Heurística
4.1. Prevención de errores	¿El Mashup puede prevenir los errores que se puedan dar a lo largo de su ejecución?
4.2. Recuperación de errores	¿Podría el Mashup recuperarse cuando un error ocurre?
4.3. Reemplazo o aviso de componentes con problemas	¿Puede el Mashup reemplazar un componente por uno más adecuado cuando este no está disponible?

4.5. Estética de Interfaces de Usuario

La estética de interfaces de usuario hace referencia a aquellos aspectos que influyen en cómo el usuario percibe la página desde su punto de vista estético, a pesar de que esta característica resulta un tanto subjetiva al gusto de cada persona o condicionada por muchos factores en un contexto determinado de uso, es posible definir atributos presentes que puedan ser generales y puedan tener un impacto general para la mayor parte de usuarios.

En este caso no se han agregado atributos específicos para Mashups, ya que esta sub-característica presenta la información siguiendo las pautas de los modelos de calidad base, a continuación presentamos una breve explicación de los atributos, junto con la tabla que muestra los atributos que están contenidos dentro de la estética de interfaces de usuario.

Esta sub-característica se divide en la uniformidad del color de fondo, haciendo referencia a si es que el Mashup presenta un color de fondo uniforme a lo largo de su ejecución, cabe destacar que generalmente el Mashup se ejecuta en una sola página Web, por lo cual esta característica más tiene que ver con que si a lo largo de la ejecución del Mashup el color de fondo se mantiene y conserva su uniformidad. Luego encontramos el color, tamaño y forma de letra y si esta está aplicada de manera uniforme en las secciones del Mashup, conservando una coherencia lógica, es decir siendo aplicado de la misma manera en todas las secciones iguales del Mashup. Consistencia en diseño hace alusión a que si está el mismo diseño aplicado a través de la interface de usuario a lo largo de toda la ejecución del Mashup. Adicionalmente, se tiene la interface, apariencia y personalización que va relacionado a que si se dan opciones de personalización de la apariencia de la interface de usuario. Por ejemplo para usuarios avanzados, normal, características favoritas, temas y colores. Con respecto al grado de

interactividad se habla de que si la información presentada necesita alguna interacción del usuario a fin de que éste provea nuevas perspectivas en la visualización de los datos.

Atributos	Explicación/Heurística
5.1. Uniformidad en el color de fondo	¿El Mashup tiene un color de fondo uniforme?
5.2. Font color/size/face uniformity	¿Está el mismo color de fuente/tamaño/apariencia aplicado en las mismas secciones?
5.3. Consistencia en el diseño	¿Está el mismo diseño aplicado a través de la interface de usuario?
5.4. Interface, apariencia, personalización.	¿Se dan opciones de personalización de la apariencia de la interface de usuario? (avanzado/normal UI, características favoritas, temas, colores)
5.5. Grado de interactividad	¿Necesita la información presentada la interacción del usuario para proveer nuevas perspectivas en la visualización de los datos?

Tabla 4-4 Sub-característica Estética de interfaces de usuario.

4.6. Accesibilidad

Con respecto a la accesibilidad y al igual que en las secciones anteriores, se ha tomado de base el trabajo de las aplicaciones Web en general, teniendo aquí ítems tales como el apoyo de lupa que hace referencia a si se proveen mecanismos para incrementar o disminuir los elementos multimedia (textos, imágenes, etc.). El soporte de texto alternativo, que controlaría si se provee un texto alternativo para describir y etiquetar los elementos multimedia. Los colores saludables, es decir si se considera que los colores empleados no dañan la integridad de los usuarios con problemas específicos. El grado de conformidad con WCAG, es decir si tiene capacidad la aplicación de seguir las directrices del Web Content Accessibility Guidelines.

Atributos	Explicación/Heurística
6.1. Apoyo de lupa	Se proveen mecanismos para incrementar / disminuir los elementos multimedia (textos, imágenes, etc.)?
6.2. Soporte de texto alternativo	¿Se provee un texto alternativo para describir y etiquetar los elementos multimedia?
6.4. Colores saludables	¿Se considera que los colores empleados no dañan la integridad de los usuarios con problemas específicos?
6.5. Grado de conformidad con WCAG	¿Tiene capacidad la aplicación Web de seguir las directrices del Web Content Accessibility Guidelines?

Tabla 4-5 Sub-característica Accesibilidad.

4.7. Conformidad

La conformidad se refiere a cómo la aplicación Web es consistente con respecto a reglas, estándares, convenciones y guías de diseño empleadas en el dominio Web.

En el modelo de usabilidad base estas sub-características fueron descompuestas en otras sub-características tales como el grado de cumplimiento con el ISO /IEC 25000 SQuaRE (2005) desde que este estándar está basado en el modelo y el grado con el que se cumplen algunas de las más relevantes guías concernientes a la usabilidad y el diseño Web. Estos atributos pueden ser cuantificados chequeando que porcentaje de patrones o guías propuestas en esos estándares han sido consideradas en el desarrollo de la aplicación Web. La tabla a continuación muestra más detalladamente lo expuesto:

Atributo
7.1. Grado de cumplimiento con el estándar ISO / IEC 25000 SQuaRE (2010)
7.2. Grado de cumplimiento con el "Research-Based Web Design & Usability Guidelines" (2006)
7.3. Grado de cumplimiento con el "Web Style Guide" (2002)
7.4. Grado de cumplimiento con el "Microsoft Web Design Guidelines" (2009)
7.5. Grado de cumplimiento con el "Sun Guide to Web Style" (2009)
7.6. Grado de cumplimiento con el "IBM Web Design Guidelines" (2009)

Capítulo 5. Método de Evaluación de la Usabilidad de Mashups

En este apartado diseñaremos el proceso de la evaluación y qué información será recolectada durante el mismo. Para poner en práctica el método de evaluación propuesto, se empleará como artefacto de entrada nuestro modelo de usabilidad definido en el capítulo 4, el mismo que, como se ha comentado en secciones anteriores, tiene como base el modelo propuesto en [19].

Los sistemas computacionales y productos de software se usan para ejecutar una gran variedad de funciones tanto de negocio como personales. Los productos de software y sistemas de computación de alta calidad, son esenciales para proveer valor y evitar consecuencias negativas potenciales para los stakeholders.

Cuando se habla de stakeholders se incluyen aquellos quienes desarrollan, adquieren, usan o son clientes de negocios del producto de software implicado.

Es importante que las características de calidad sean especificadas, medidas y evaluadas siempre que sea posible, usando métricas validadas o ampliamente aceptadas y métodos de medición. Los modelos de calidad en el SQuaRE, pueden ser usados para identificar características de calidad relevantes, que pueden ser usadas para establecer requisitos, su criterio o satisfacción y las medidas correspondientes.

Como se mencionó anteriormente, en este capítulo se presenta un método para la evaluación de la calidad, tomando de base el estándar SQuaRE, específicamente su división ISO 2504n en su versión 3 lanzada el 14 de Junio de 2009, que consiste de los siguientes estándares:

- **Estándar ISO / IEC 25040 – Modelo de referencia de evaluación y guía:** contiene requisitos para la especificación y evaluación de la calidad del software y clarifica los conceptos generales. Provee una descripción del proceso para evaluar calidad del producto software y ubica los requisitos para la aplicación de este proceso. El proceso de evaluación es la base para la evaluación de la calidad del producto de software para diferentes propósitos y enfoques. De ahí el proceso puede ser usado para la evaluación de la calidad en uso, medida externa de calidad de software y medida interna de calidad de software; como también, puede ser aplicado para evaluar la calidad de software pre-desarrollado o software personalizado durante su proceso de desarrollo. La evaluación de la calidad del producto de software puede ser conducida, por ejemplo por un adquiriente, una organización desarrolladora o un evaluador independiente.
- **Estándar ISO / IEC 25041 – Módulos de evaluación:** define la estructura y contenido de la documentación a ser usada para describir un módulo de evaluación. Estos módulos de evaluación, contienen la especificación del modelo de calidad (por ejemplo características, subcaracterísticas y las correspondientes métricas de calidad interna, externa o en uso), los datos asociados y la información sobre la aplicación planeada del modelo y de la información sobre su aplicación actual. Los módulos de

evaluación apropiados son seleccionados para cada evaluación. En algunos casos puede ser necesario desarrollar nuevos módulos de evaluación. La guía para desarrollar nuevos módulos de evaluación es encontrada en la ISO / IEC 25041.

- **Estándar ISO / IEC 25045 – Módulos de evaluación para la recuperación:** Provee la especificación para evaluar las sub-características de recuperación definidas bajo la característica de confiabilidad del modelo de calidad. Este determina las medidas externas de la calidad de software de resistencia y del índice de recuperación autónoma cuando el sistema de información esté compuesto de uno o más productos de software, la ejecución de las transacciones está sujeta a una serie de distracciones. Una distracción puede ser un fallo operacional (por ejemplo un apagado abrupto o un proceso de sistema operativo que haga caer al sistema) o un evento (por ejemplo un incremento significativo de usuarios al sistema).

5.1. Definición del proceso con SPEM 2

En esta sección se detalla cómo en base a lo anteriormente explicado se llevará a cabo el proceso de evaluación correspondiente a la aplicación de la metodología de evaluación, para ello usaremos la notación SPEM 2 (Software & System Process Engineering Meta-model Specification V2.0)[51], propuesta por el grupo Object Management Group, que provee un marco formal para la definición de procesos de desarrollo de sistemas y de software, así como también para definir y describir sus elementos. El objetivo de su uso en este trabajo, es el proveer una definición detallada del proceso de evaluación de la usabilidad de los Mashups y guiar al evaluador para llevar a cabo los procesos de una manera simple y fácil de entender.

SPEM 2 se ubica de la Ingeniería de Procesos de Software (SEP), dedicándose a la definición, implementación, medición y mejora de los procesos de Ingeniería del Software. Está basada en MOF (Meta Object Facility), que es un estándar de la OMG, siendo SPEM a los procesos software lo mismo que UML a los sistemas software.

SPEM es un meta-modelo que permite definir modelos de procesos de Ingeniería del Software y de Ingeniería de sistemas. Este detalla los elementos mínimos necesarios que permitan definir dichos procesos sin añadir aspectos del dominio particular. La idea central de SPEM 2 está basada en tres elementos básicos: rol, producto de trabajo y tarea.

Tarea	Esfuerzo a realizar.
Rol	Quien realiza el esfuerzo.
Productos de trabajo:	Entradas que se utilizan y salidas que estos producen.

Por medio de estos elementos se puede especificar “Quien (rol) realiza qué (tarea) para desde unas entradas conseguir unas salidas (productos de trabajo)”.

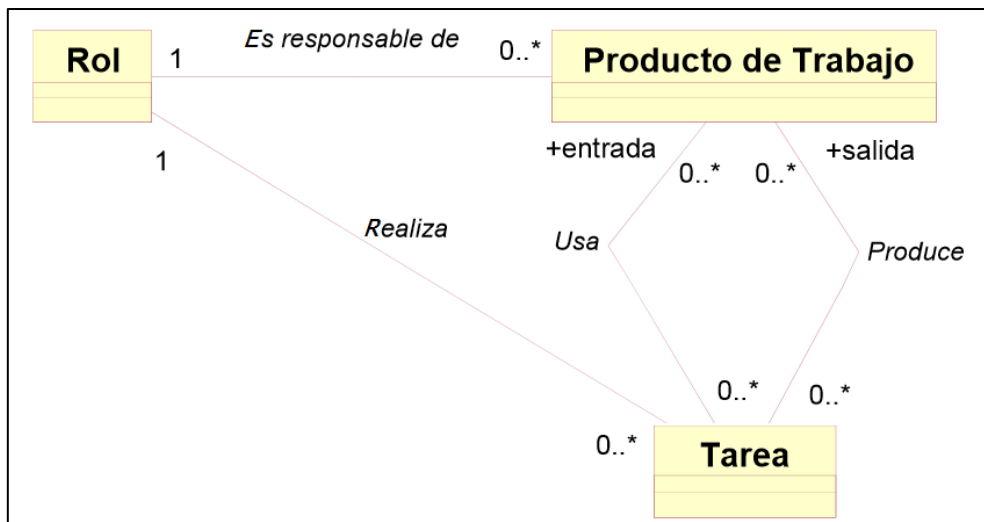


Figura 5-1 Metamodelo SPEM

SPEM 2 distingue dos etapas cuando se define un proceso:

1. Definición de los elementos de contenido que son los elementos primarios o constructores básicos.
 - a. Roles
 - b. Tareas.
 - c. Productos de trabajo.
 - d. Categorías.
2. Luego se combinan y reutilizan los elementos esenciales para obtener los Procesos.

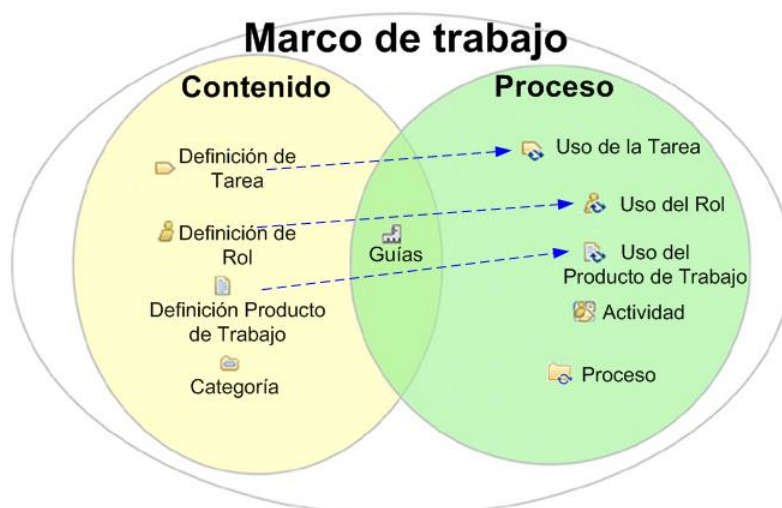


Figura 5-2 Mapeo contenido-proceso SPEM 2.

A continuación se describe el subconjunto de primitivas de modelado que son las más comúnmente utilizadas a la hora de definir un proceso:













Icono	Nombre	Descripción
	Definición de Rol	Conjunto de habilidades, competencias y responsabilidades relacionadas, de un individuo o de un grupo.
	Definición de Tarea	Unidad de trabajo asignable y gestionable, identificando el trabajo que se ejecuta por los roles. Puede dividirse en varios pasos.
	Definición de Producto de trabajo	Producto usado o producido por las <i>Tareas</i> . Existen dos tipos de productos: <i>Artefacto</i> de naturaleza tangible (modelo, documento, código, archivos, etc.) y <i>Entregable</i> para empaquetar productos con fines de entrega a un cliente interno o externo. Se pueden asociar entre ellos mediante relaciones de agregación, composición e impacto.
	Categoría	Clasificación de elementos como <i>Tareas</i> , <i>Roles</i> y <i>Productos</i> en base a los criterios que desee el ingeniero de procesos. Existen diversos tipos de categorías: <i>Conjunto de Roles</i> (para <i>Roles</i>), <i>Disciplina</i> (para <i>Tareas</i>), <i>Dominio</i> (para <i>Productos</i>),
	Guías	Información adicional relacionada con otros elementos. Los sub-tipos de guías pueden ser (entre otros): <i>Activo Reutilizable</i> , <i>Directriz</i> , <i>Documentación</i> , <i>plantillas</i> . El icono presentado es genérico, pero se pueden emplear
	Uso de Rol	Representación del <i>rol</i> que lleva a cabo una <i>Tarea</i> o <i>Actividad</i> dentro de un proceso determinado. Hace referencia a una Definición de Rol (elemento de Contenido).
	Uso de Tarea	Representación de una <i>tarea</i> atómica dentro de un proceso determinado. Hace referencia a una Definición de Tarea (elemento de Contenido).
	Uso de Producto de Trabajo	Representación de un <i>Producto de Trabajo</i> de entrada o salida, relacionado con una <i>Actividad</i> o <i>Tarea</i> . Hace referencia a una Definición de un Producto de Trabajo (elemento de Contenido)
	Actividad	Representación de un conjunto de <i>Tareas</i> que se ejecutan dentro del proceso, junto con sus <i>Roles</i> y <i>Productos</i> asociados. Si únicamente se quiere representar una agrupación de tareas, se puede usar los elemento Actividad o Fase (incluido por retro-compatibilidad y más empleado en tareas de desarrollo), o bien si es un conjunto de tareas que se repite un determinado número de veces, se puede usar el elemento Iteración.
	Fase	
	Iteración	
	Paquete de Proceso	Representación de un paquete agrupando todos los elementos del proceso

Tabla 5-1 Nomenclatura SPEM 2.

Ventajas de utilizar SPEM:

- Se puede disponer de modelos de Procesos de software en formato procesable por computador.
- Facilita la comprensión y comunicación entre las personas puesto que propicia un conocimiento homogéneo.
- Da soporte a la mejora de procesos.
- Da soporte a la gestión de procesos.
- Guía la automatización de procesos y da soporte para la ejecución automática.

5.2. Método de Evaluación con SPEM 2

El primer paso para definir la aplicación del modelo de evaluación será especificar el proceso de construcción de un Mashup. La figura a continuación define un proceso de composición que refleja los estados que se suelen incorporar en el desarrollo de los Mashups [12]. Aquí los estados son representados mediante “Tareas” sin entrar en detalle, ya que el objetivo es tener una noción de los roles que realizan tales fases junto con los productos consumidos y generados.

En el método de evaluación aquí planteado, no se tendrá en cuenta el proceso de construcción de componentes, se asumirá que se parte de un conjunto finito de componentes publicados en Internet.

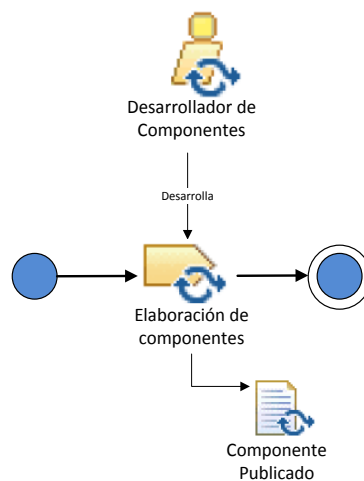


Figura 5-3 Proceso de construcción de un componente

A continuación se describe en SPEM un proceso genérico del ciclo de vida del desarrollo de los Mashups.

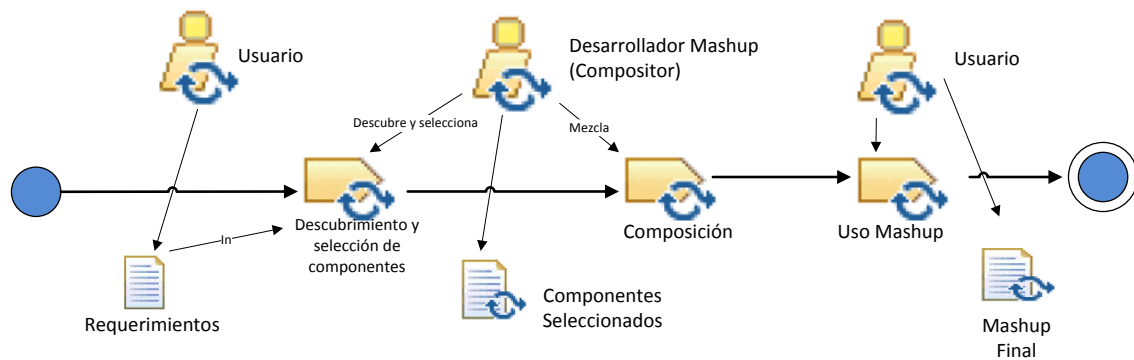


Figura 5-4 Diagrama del ciclo de vida de construcción de un Mashup con SPEM.

Las fases del proceso de desarrollo de los Web Mashups propuestos en [12], se basan en las actividades relacionadas al desarrollo del Mashup, desde la construcción del componente (no abordado como parte de este trabajo), su puesta a disposición en la Web para que quien compone el Mashup se encargue de la selección de los componentes, el proceso de composición del Mashup y luego el posterior uso por parte del usuario final de la aplicación Mashup.

Antes de ejecutar todas estas tareas propias del Mashup, se debería conocer los requisitos de la aplicación, ya que esto es necesario para tener claro lo que se espera del producto final y los requisitos que se quieren satisfacer, teniendo en cuenta que un Mashup generalmente es una aplicación creada para solventar una necesidad específica, por lo general tiene la extensión de una página Web, muchas veces construida por el mismo usuario final por lo que se puede mezclar el rol del desarrollador con el usuario, cabe señalar también que aquí los requisitos se conocerían de antemano ya que al ser una aplicación para cubrir una necesidad puntual y de corta duración, no se habla de un sistema extenso, que requiera una elicitación profunda de requisitos, sino más bien de un conocimiento claro del dominio y de tener el objetivo que se persigue con la construcción de la aplicación bien establecido.

Según [12], se tiene como rol el del Desarrollador de Componentes, cuyo objetivo es crear un nuevo componente que cumpla con aspectos funcionales y no funcionales, sin embargo aquí no lo abordaremos ya que en este trabajo se toma únicamente en cuenta la construcción del Mashup como producto de Software y los componentes como a cajas negras en las cuales son consumidos como productos intermedios, sin importar sus características individuales sino el funcionamiento y características no funcionales una vez que han pasado a formar parte del Mashup.

El desarrollador o compositor integra los componentes dentro del Mashup. El compositor típicamente busca en la Web, componentes que reúnan las características funcionales y no funcionales requeridas por el usuario para el Mashup. Esto es, el compositor descubre los componentes y los selecciona. Haciendo esto, toma en consideración no solamente sus propias necesidades (por ejemplo que sea fácil de componer), sino las necesidades y expectativas del usuario final.

El proceso de evaluación de usabilidad de los Mashups, ha sido definido tomando como punto de partida el proceso de evaluación de calidad propuesto en el estándar 2504n,

teniendo en cuenta las fases del mismo, éstas se han adaptado y extendido de acuerdo a las características del producto aquí estudiado.

La división 2504n de la norma SQuaRE, propone un método de evaluación de la calidad basado en cinco fases principales detalladas en el siguiente gráfico:

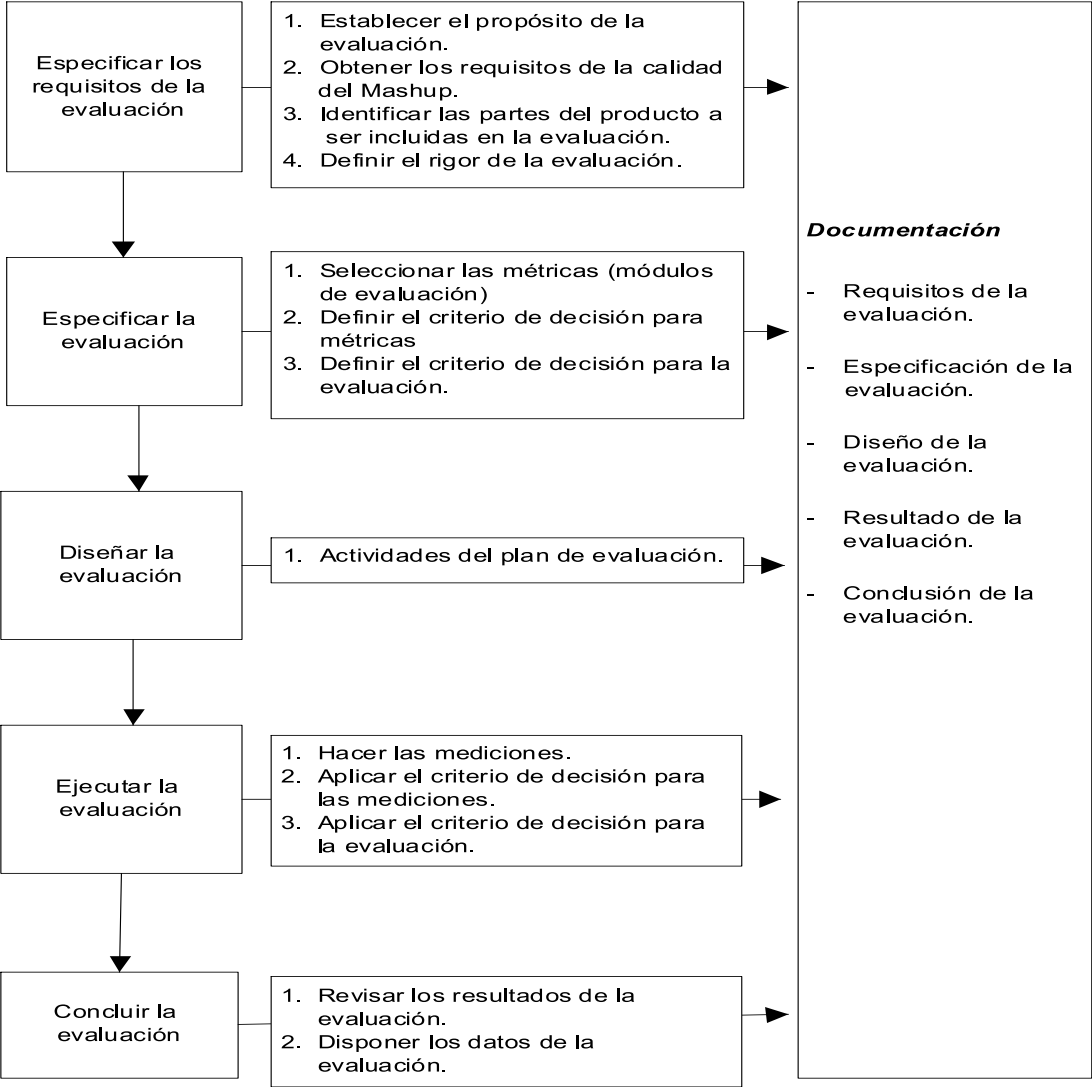


Figura 5-5 Pasos del Proceso de Evaluación de la Usabilidad de Mashups

Para el caso de los Mashups, hemos adaptado y extendido cada una de las tareas que se realizan en las fases anteriores para la evaluación de la usabilidad de los mismos, la figura a continuación, muestra una vista general de las principales fases que tienen lugar en el proceso.

A continuación se presenta cada una de las fases antes descritas con sus respectivas tareas y su detalle.

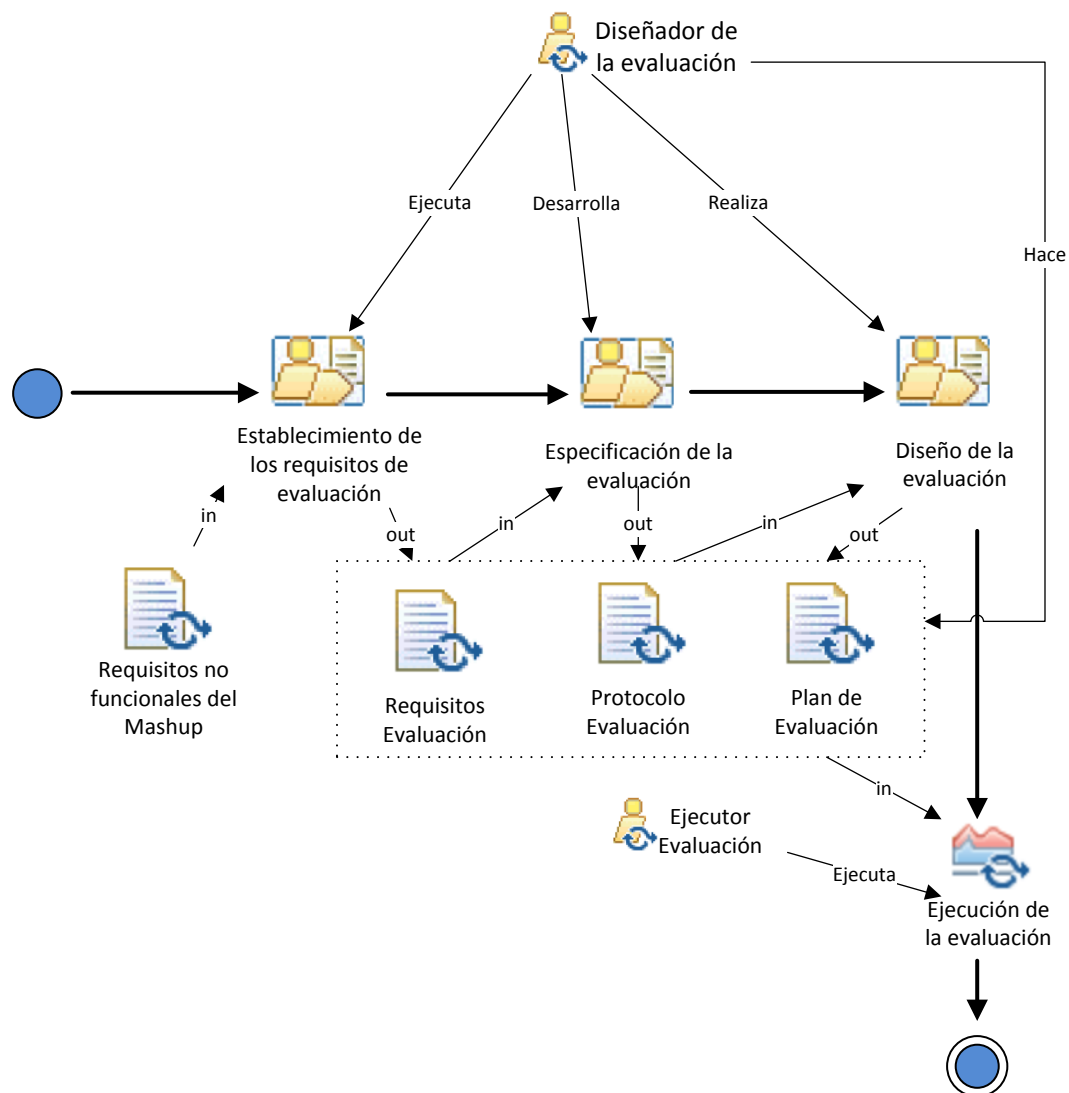


Figura 5-6 Proceso de evaluación de usabilidad para Mashups

5.2.1. Especificación de los Requisitos de Evaluación

En esta fase se especifican los requisitos que se tendrán en cuenta al momento de diseñar la evaluación de la usabilidad de los Mashups. El objetivo es centrar el enfoque de la evaluación de la usabilidad atendiendo a todos los factores que condicionan e Mashup.

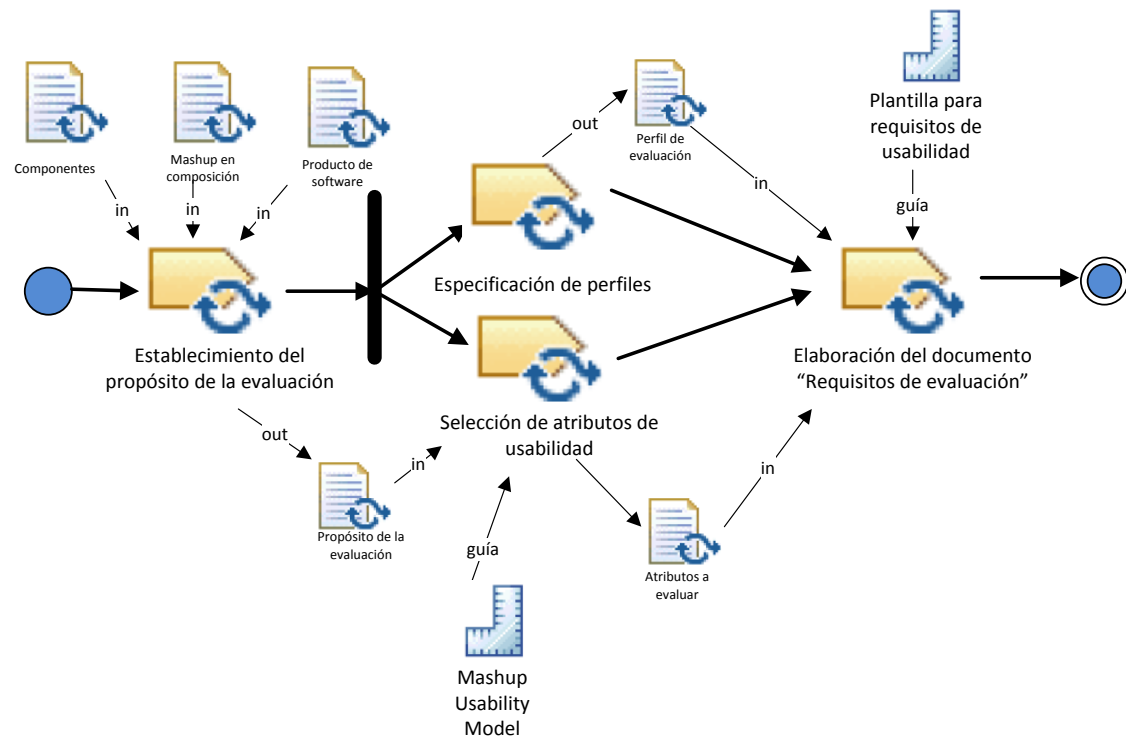


Figura 5-7 Especificación de los “Requisitos de Evaluación”

a) Establecimiento del Propósito de la Evaluación de la Usabilidad del Mashup

Esta tarea consiste en determinar en qué momento se realizará la evaluación de la calidad del Mashup:

- Mientras se realiza la selección de los componentes (evaluación de usabilidad formativa) si se pretende realizar la evaluación de usabilidad como una proyección de la usabilidad del Mashup, es decir si se tomará cada uno de los componentes como cajas negras y se definirá la calidad en cuanto a usabilidad que cada uno de éstos aportará al Mashup.
- Mientras se realiza el proceso de composición (evaluación de usabilidad formativa) si se pretende evaluar la usabilidad que se obtendrá del proceso de unir los componentes seleccionados, proyectando su comportamiento conjunto y analizando la usabilidad de la composición.
- Cuando el Mashup está ya siendo usado el usuario final en un determinado contexto de uso (evaluación de usabilidad sumativa) si se pretende evaluar la usabilidad en un contexto de uso determinado y bajo factores específicos.
- También se podría realizar una evaluación completa en la cual el proceso de desarrollo del Mashup está siendo controlado siempre y compara las predicciones de usabilidad con los resultados obtenidos a través del testeo con usuarios.

b) Especificación de Requisitos de Usabilidad del Mashup

Esta tarea consiste en especificar qué requisitos de calidad debe cumplir la evaluación, así como también quien los solicita.

- El evaluador debe asegurar que los stakeholders del Mashup sean identificados. La información sobre quien requiere la evaluación puede ser necesaria para identificar todos los stakeholders; los mismos que pueden ser una persona, parte u organización y pueden estar involucrados en la evaluación. Dos clases de stakeholders pueden ser identificados. Uno es un stakeholder del Mashup, tal como el desarrollador, adquiriente, usuario, etc. Otro es aquel que requiere la evaluación quien necesita información sobre la calidad de software, patrocina la evaluación y requiere un reporte de la evaluación.
- Empleando el Modelo de Usabilidad de Mashups descrito en secciones anteriores, como un artefacto de entrada a modo de catálogo, se seleccionarán las características, sub-características y atributos que se van a evaluar.

c) Elaboración del Documento de Requisitos de Evaluación

Una vez reunida toda la información anteriormente expuesta, se elaborará un documento que refleje los requisitos de la evaluación. Para ello se pueden usar plantillas, las mismas que contendrán la información de las tareas precedentes. En el Anexo C, encontramos la Plantilla 1 Documento de Especificación de Requisitos de Evaluación.

5.2.2. Especificación de la Evaluación

En esta fase se especifica detalladamente la evaluación que se realizará, qué artefactos se van a evaluar, qué métricas se aplicarán y de qué manera, qué umbrales se considerarán adecuados para los resultados obtenidos, cómo se combinan los resultados y cómo se reportan los problemas que sean detectados.

Esta fase consta de las tareas que se muestran en la figura y se detallan a continuación:

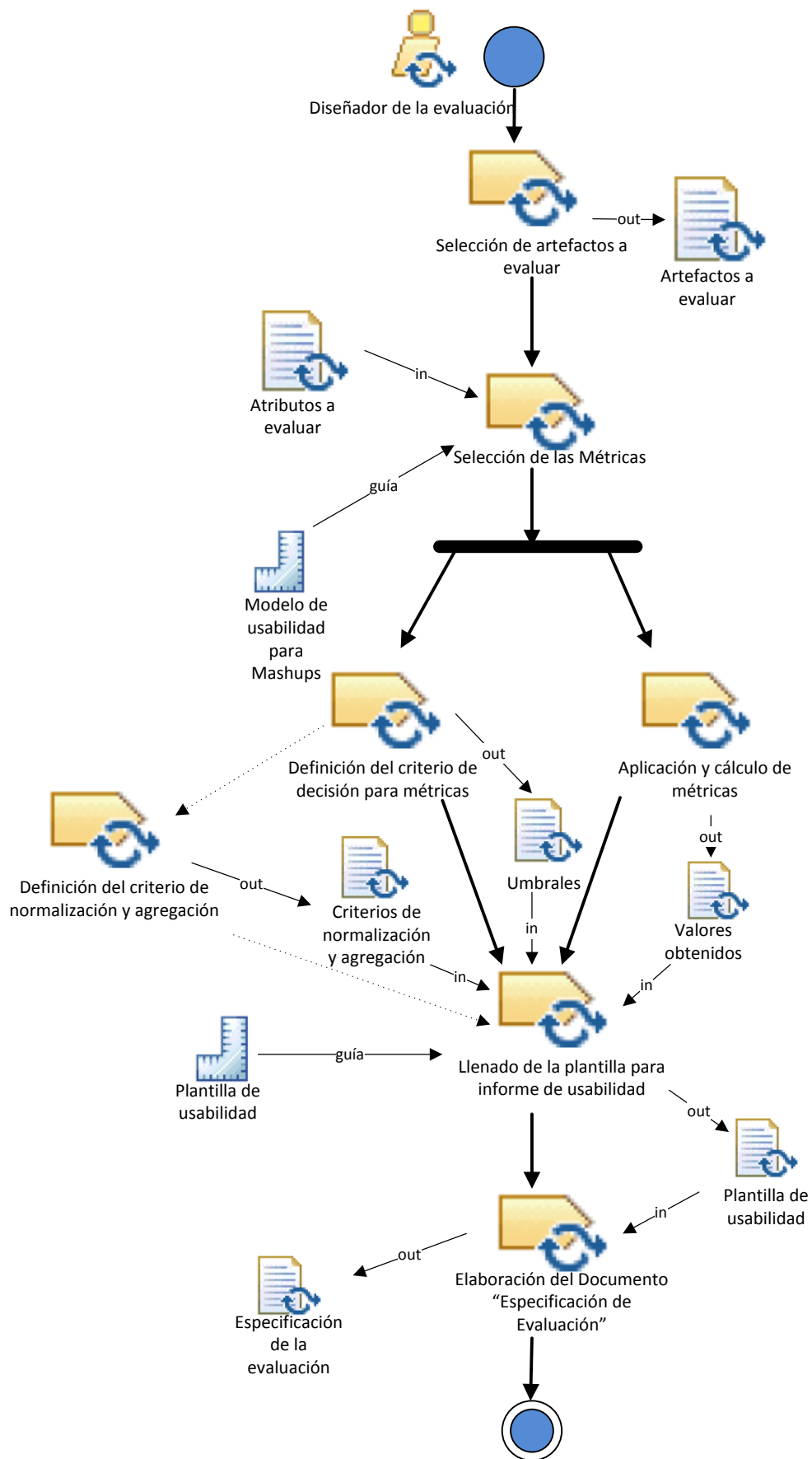


Figura 5-8 *Especificación de la evaluación*

a) Selección de Artefactos a Evaluar

Los artefactos a evaluar pueden ser

- Los productos intermedios, en este caso los componentes individualmente previa la composición, considerándolos como cajas negras, es decir sin importar los detalles de la evaluación.
- Los componentes cuando están interactuando entre sí, es decir cuando forman ya parte de un Mashup.
- El Mashup como aplicación Web terminada.

Los métodos de evaluación deberán ser documentados, teniendo en cuenta las acciones a ser ejecutadas para conseguir los resultados de la evaluación. Cuando el método de evaluación descrito está basado en el uso de una herramienta de software, esta herramienta deberá ser identificada en el plan de evaluación. Tal identificación debe incluir al menos el nombre de la herramienta, la identificación de su versión y su origen.

b) Selección de Métricas a Emplear

Una vez fijados los atributos de usabilidad que serán abordados dependiendo de los artefactos a evaluar, podemos emplear el modelo de usabilidad de Aplicaciones Web Mashup definido anteriormente, con el fin de establecer las métricas asociadas a los atributos seleccionados. Una vez que se han obtenido las métricas, se debe consultar su definición genérica para conocer qué aspectos se pretende medir y a que niveles de abstracción se pueden evaluar.

c) Definición del Criterio de Decisión para las Métricas de Usabilidad

El criterio de decisión deberá ser establecido para las métricas seleccionadas.

Los criterios de decisión son umbrales numéricos u objetivos usados para determinar la necesidad de acciones futuras o investigación, para describir el nivel de confianza en un resultado dado. Para aquellas métricas que han sido vinculadas a los artefactos a evaluar, se ha de establecer un mapeo entre la descripción que plantea la métrica y los elementos propios del artefacto a evaluar. Aquí se obtendrá una fórmula para calcular el estado del artefacto a evaluar.

Esta tarea es relevante para realizar una evaluación cualitativa. Para cada métrica se establecen los intervalos de valores que representan el grado de usabilidad del Mashup y que permiten tomar posteriores decisiones. Gracias a estos intervalos de valores se podrán detectar qué problemas de usabilidad existen y clasificarlos de acuerdo a su nivel de criticidad. Para ello se toma cada uno de los intervalos y se asigna un indicador. Por ejemplo, si el rango de valores que ofrece una métrica puede resultar en valores que van desde 0 a 1 en donde 1 es el valor óptimo a alcanzar para el grado más alto de usabilidad, se establecen umbrales de tal forma que el valor obtenido "X" se enmarque en uno de ellos, si $[0 \leq x \leq 0.5]$ podría por ejemplo tener un problema grave de usabilidad, si estuviera entre $[0.5 \leq x \leq 0.75]$ existiría un problema

de usabilidad medio y si está entre $[0.75 \leq x \leq 0.9]$ podría ser un problema de usabilidad leve, no encontrándose problemas de usabilidad entre el rango $[0.9 \leq x \leq 1]$.

d) Definición del Criterio de Normalización y Agregación

Esta tarea consiste en establecer una evaluación cuantitativa, en el caso de los Mashups puede ser utilizado por ejemplo en la fase de selección de componentes con la finalidad de escoger las mejores opciones en base a ciertos atributos de usabilidad que se quieren obtener. Para cada sub característica y atributo seleccionado del Modelo de Usabilidad Web se puede definir una fórmula que combine los valores obtenidos para así obtener un indicador que cuantifique el grado de contribución a la usabilidad del componente. En el caso de atributos con más de una métrica asociada, se pueden emplear expresiones que asignen pesos para ponderar los valores obtenidos o reglas para la combinación de valores no numéricos. Se podría requerir la normalización previa de valores si sus tipos de escalas son distintas. Este tipo de proceso se puede seguir recursivamente con las subcaracterísticas que tengan diversos atributos asociados.

e) Definición de la Plantilla para el Informe de Usabilidad

Esta tarea consiste en diseñar un documento que presente los problemas de usabilidad detectados para así emprender acciones correctivas. Por tanto, un informe de usabilidad es una lista de problemas de usabilidad detectados. Para cada problema de usabilidad se deben incluir los siguientes campos:

- Identificador del Problema: Código que identifica al problema de forma única.
- Descripción: Detalle textual del problema detectado.
- Atributo afectado: Atributo de usabilidad que se ve afectado por el problema en cuestión, se debe indicar la proveniencia del atributo.
- Nivel de criticidad: Qué tipo de problema es, si este es leve, medio o grave.
- Artefacto evaluado: Se debe explicar de qué atributo proviene el problema, es decir que artefacto fue evaluado, si fue un componente, la mezcla de ellos o el mashup final.
- Origen del problema: Hace referencia a los artefactos que originan dicho problema, éste no necesariamente puede ser el artefacto evaluado en conjunto, por ejemplo si estamos midiendo la usabilidad de un mashup, no necesariamente originará el problema la mezcla de componentes o el Mashup final, el problema pudo haberse generado en un componente en particular.
- Ocurrencias: Número de veces en las que el problema de usabilidad se repite.
- Recomendaciones: Sugerencias dadas por el evaluador con fines correctivos y/o preventivos.
- Prioridad: Importancia que se le otorga al problema a la hora de ser corregido.

Además puede ser de interés incorporar información sobre el impacto que tiene el problema en la usabilidad, el esfuerzo que la corrección del problema implica y los cambios que se realizarán basándose en la prioridad, el esfuerzo, el impacto, etc.

Teniendo en cuenta que los Mashups son Aplicaciones Web, se ha tomado de base el trabajo de [19] en donde se define una plantilla de evaluación para procesos de evaluación de usabilidad en aplicaciones Web, por tanto el siguiente formato será utilizado:

ID	XXX Código que identifica de forma única al problema de usabilidad en cuestión.
Descripción	Descripción textual del problema detectado en base al resultado obtenido de la métrica y los elementos que participan en el cálculo de la misma
Atributo Afectado	ID. Sub-característica / ... / ID. Atributo. Atributo de usabilidad perteneciente al Modelo de Usabilidad de Mashups que se ve afectado por este problema, indicando las sub-características de las que proviene.
Nivel de Criticidad	[Mayor / Medio / Menor] (resultado de la métrica) Nivel de criticidad según los intervalos definidos anteriormente para las métricas. Si el problema existe en varios de los artefactos evaluados (del mismo tipo y nivel de abstracción), se considera como nivel de criticidad el más alto de ellos.
Artefacto(s) evaluado(s)	Para el caso de los Mashups se tiene ya sea el Mashup en su estado final, un conjunto de componentes a ser mezclados o simplemente un componente tomado como una caja negra.
Ocurrencias	Número de veces que el problema de usabilidad se repite en cada artefacto evaluado.
Recomendaciones	Sugerencias que permiten prevenir y/o corregir dichos problemas
Prioridad	[Alta, Media, Baja] Prioridad a la hora de realizar los cambios propuestos en el campo recomendaciones (a rellenar por el desarrollador responsable del artefacto a modificar).
Recursos	Recursos (temporales, económicos, etc.) o la falta de ellos para realizar los cambios propuestos (a rellenar por el desarrollador responsable del artefacto a modificar).

5.2.3. Diseño de la Evaluación

Dentro de esta fase se elabora un plan de evaluación con toda la información necesaria ya obtenida de fases anteriores, en el plan de evaluación se detallan aspectos técnicos sobre la evaluación. Aquí las actividades de la evaluación de la calidad del Mashup deben ser programadas, teniendo en cuenta la disponibilidad de los recursos tales como personal que intervendrá en la evaluación, herramientas de software y computadores.

Como la naturaleza de la evaluación es iterativa, cuestiones pueden ser revisadas según la información obtenida.

La evaluación de los Mashups puede ser llevada a cabo en etapas, en diferentes puntos en el desarrollo de un ciclo de vida, o todas a la vez, a un punto del ciclo de vida. Para el caso

específico de los Mashups, se deberá tener en cuenta que el número de individuos involucrados es generalmente limitado al desarrollador y al usuario final, que inclusive muchas veces se trata de la misma persona.

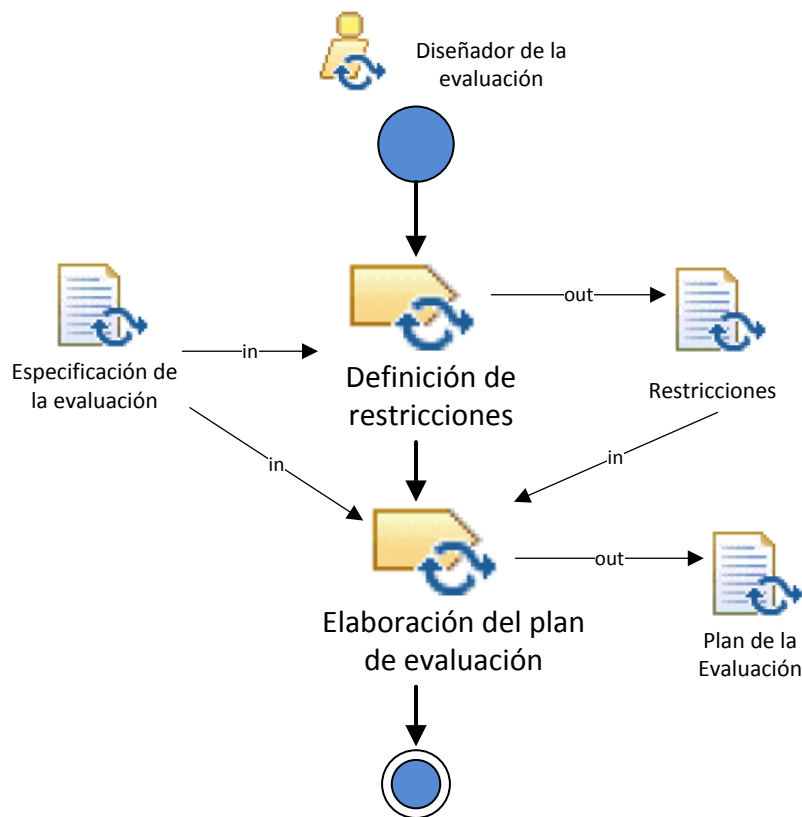


Figura 5-9 *Diseño de la evaluación*

Por tanto esta fase consta de dos tareas:

a) Definición de restricciones.

Se fijan las restricciones de la evaluación como por ejemplo:

- El presupuesto de la evaluación.
- Los métodos de evaluación y estándares adaptados.
- Las herramientas de evaluación.
- Las actividades de evaluación, incluyendo el itinerario y recursos involucrados.
- Las restricciones en cuanto al acceso a productos intermedios en este caso por ejemplo la restricción al acceso a alguna característica de cierto componente.

b) Elaboración del plan de evaluación.

En base a todos los artefactos que se pretende evaluar, se establece un orden de evaluación y se asignan tareas a cada una de las personas encargadas de poner en ejecución la evaluación.

El plan de evaluación debe evitar tareas duplicadas con la evaluación y definir puntos de decisión en el proceso de la evaluación el cual determina cuando y porque la

evaluación puede ser considerada completada y debería parar. Esto debería ser hecho para disminuir el riesgo de errores y para reducir el esfuerzo de evaluación planeado.

5.2.4. Ejecución de la Evaluación e Informe de Usabilidad

En esta fase se recibe la especificación detallada del plan de evaluación de la calidad así como los requisitos de calidad de la evaluación, así también se tendrán ya especificadas las métricas para evaluar los artefactos y obtener los resultados que permitan rellenar los informes de usabilidad correspondientes. Se presenta a continuación el diagrama de SPEM con la descripción gráfica de esta fase.

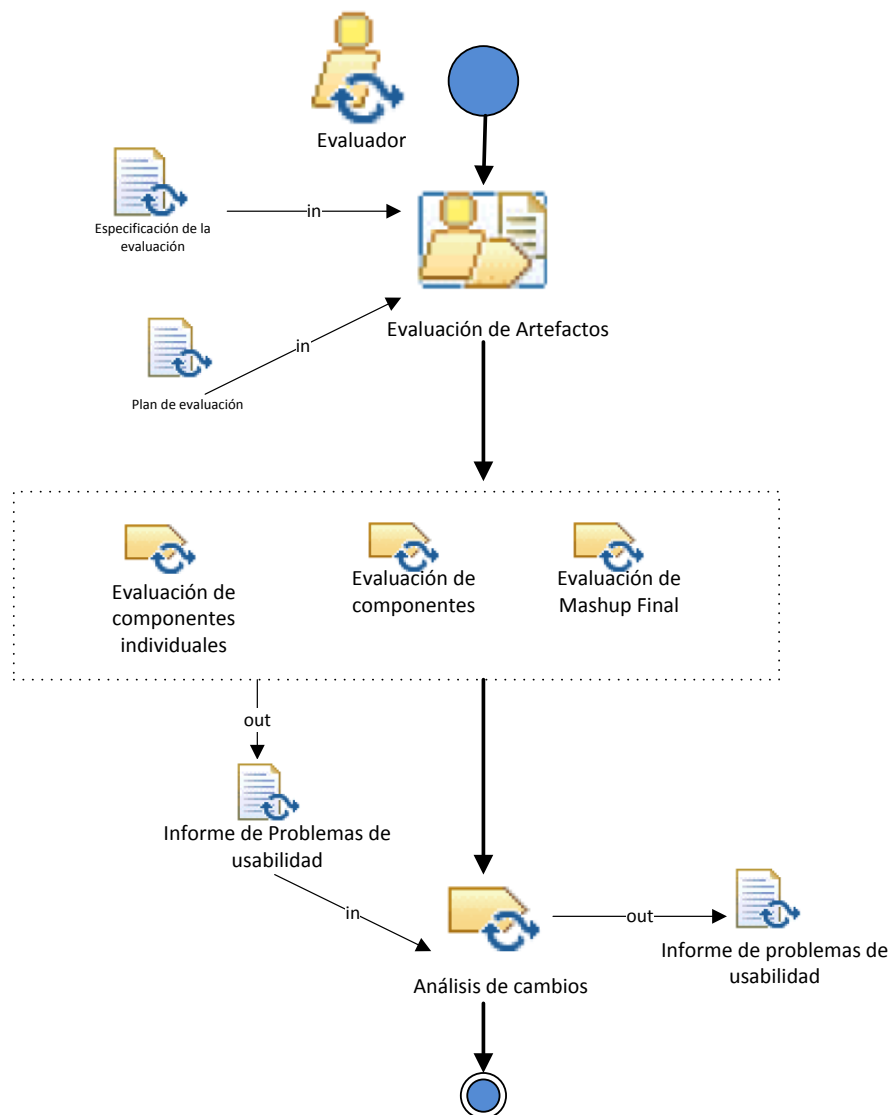


Figura 5-10 Fase "Ejecución de la evaluación"

Es importante tener en cuenta que el proceso de evaluación de artefactos es iterativo, por cuanto se puede volver a repetir el proceso de evaluación de los artefactos. En esta fase se tendrán en cuenta dos tareas principales:

a) Evaluación de los Artefactos

En esta tarea se aplican las métricas sobre los artefactos seleccionados. Una vez que los valores son obtenidos se determina el impacto y se establece el nivel de criticidad del problema de usabilidad. Con la información obtenida se puede ya elaborar el informe de usabilidad, el que consta de los problemas de usabilidad detectados y documentados conforme a la plantilla elaborada en la fase de especificación.

El criterio de decisión para valoración de la calidad del Mashup deberá ser aplicado a los valores medidos.

El resultado de la evaluación debería:

1. Establecer un apropiado grado de confianza en el que el Mashup es capaz de reunir los requerimientos de la evaluación.
2. Identificar cualquier deficiencia específica con respecto a los requerimientos de la evaluación y cualquier evaluación adicional necesaria para determinar el ámbito de esas deficiencias.
3. Identificar cualquier limitación especial o condición relacionada al uso del Mashup.
4. Identificar cualquier debilidad u omisión en la evaluación misma y cualquier evaluación adicional que sea necesaria.
5. Identificar cualquier opción para el uso del Mashup no cubierta por la evaluación.

Aquí en esta tarea por tanto, se aplica el criterio de decisión para la evaluación, siendo una evaluación sintética de la calidad del Mashup y no una valoración del proceso de software.

Las evaluaciones pueden ir desde la evaluación de los componentes como cajas negras aisladas, la evaluación de la mezcla de los componentes o la del producto final, por tanto como vemos aquí se ven implicados todos los posibles artefactos a ser evaluados.

Para cada problema, se introduce toda la información requerida en la plantilla del informe.

b) Análisis de Cambios

A partir de los informes de usabilidad obtenidos en la tarea anterior, se clasifican los problemas según su origen y se ordena según su criticidad. Esta clasificación pasará a componer los informes de problemas de usabilidad para cada una de las fases del desarrollo Web. Los problemas de usabilidad serán valorados por los desarrolladores.

c) Elaboración del Informe de Usabilidad

En la Fase de la Especificación de la Evaluación, se definió la plantilla del Informe de Usabilidad, la misma que en esta tarea dentro de esta fase, deberá ser elaborada conjuntamente con la Evaluación de los Artefactos y el Análisis de Cambios, tareas definidas en esta fase.

5.2.5. Finalización de la Evaluación

Esta actividad consiste en las siguientes tareas:

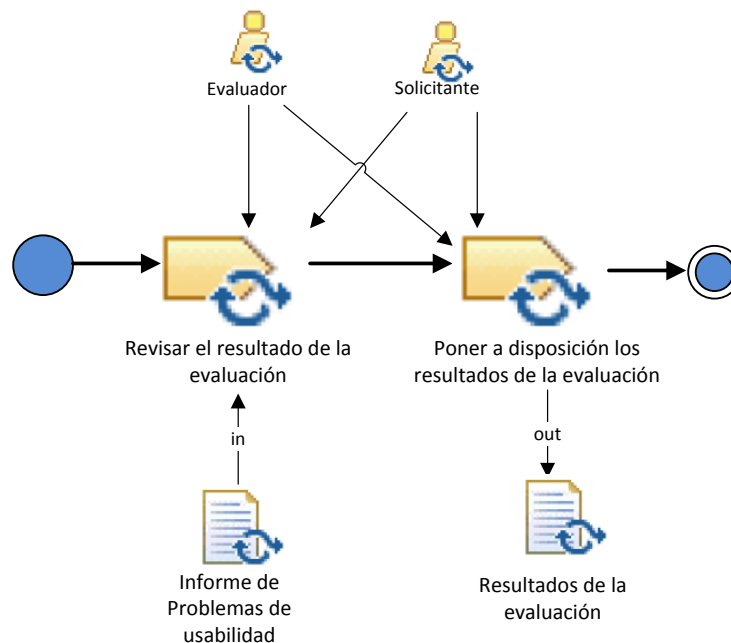


Figura 5-11 "Finalizar la evaluación"

a) Revisar el Resultado de la Evaluación

El evaluador y el que solicitó la evaluación debe llevar a cabo una revisión conjunta de los resultados de la evaluación.

Los comentarios del reporte de la evaluación deben ser dispuestos e incluidos en la versión final del reporte.

b) Disponer Datos de la Evaluación

Cuando la evaluación se ha completado, los datos y los ítems de evaluación deben ser dispuestos de acuerdo a los requerimientos del solicitante de la evaluación.

Capítulo 6. Aplicación de Método de Evaluación

En este capítulo se presenta como el método de evaluación Web aquí tratado puede ser puesto en práctica, se ha puesto a modo de caso de estudio cuatro Mashups, tomados de la lista de los más populares de Programable Web, para de esta manera proveer ejemplos que sirvan de guía para la evaluación de la calidad de los Mashups y que den una idea clara de cómo poner en práctica cada una de las tareas que se encuentran en las fases de la evaluación.

En la sección 6.1 se presenta cada uno de los Mashups a evaluar, incluyendo una breve explicación sobre su funcionalidad y qué componentes han sido utilizados para su elaboración.

En la sección 6.2 se presenta la aplicación del método de evaluación propuesto en este trabajo sobre cada uno de los Mashups anteriormente citados.

Por último en la sección 6.3 se recogen las lecciones aprendidas tras la aplicación del método de evaluación y se hacen recomendaciones basadas en la experiencia de la ejecución del método de evaluación.

6.1. Mashups a ser Evaluados

En esta sección se presentará una breve descripción sobre cada uno de los casos de estudio seleccionados para ejecutar la evaluación de la usabilidad. Se tendrá en cuenta el tipo de APIs que se han utilizado, que funcionalidad tienen y qué servicios prestan al usuario, su objetivo y finalidad.

6.1.1. Liveplasma

Dentro de la página de Programable Web, se ha tomado como primer ejemplo el Mashup www.liveplasma.com, este mashup busca música, películas y libros, y los resultados son presentados en un modelo gráfico mostrando la información relacionada al criterio de búsqueda. Este mashup combina los servicios de Amazon con YouTube.

A continuación se muestra el Mashup a evaluar y las partes que componen el mismo:



Figura 6-1 Mashup Liveplasma www.liveplasma.com

Como pantalla de bienvenida tenemos la del gráfico [anterior], en la que podemos ver la selección que queremos hacer ya sea entre música, películas o libros; conjuntamente con el motor de búsqueda según el artista si está seleccionada la opción “música”, nombre de la película, en el caso de que esté seleccionada la opción “películas” o el título del libro en el caso de seleccionar “libro”.

Para el ejemplo en este trabajo descrito, ponemos una búsqueda por “música” e ingresamos como autor por ejemplo a “Rod Stewart”, una vez realizada la búsqueda veremos la siguiente pantalla:

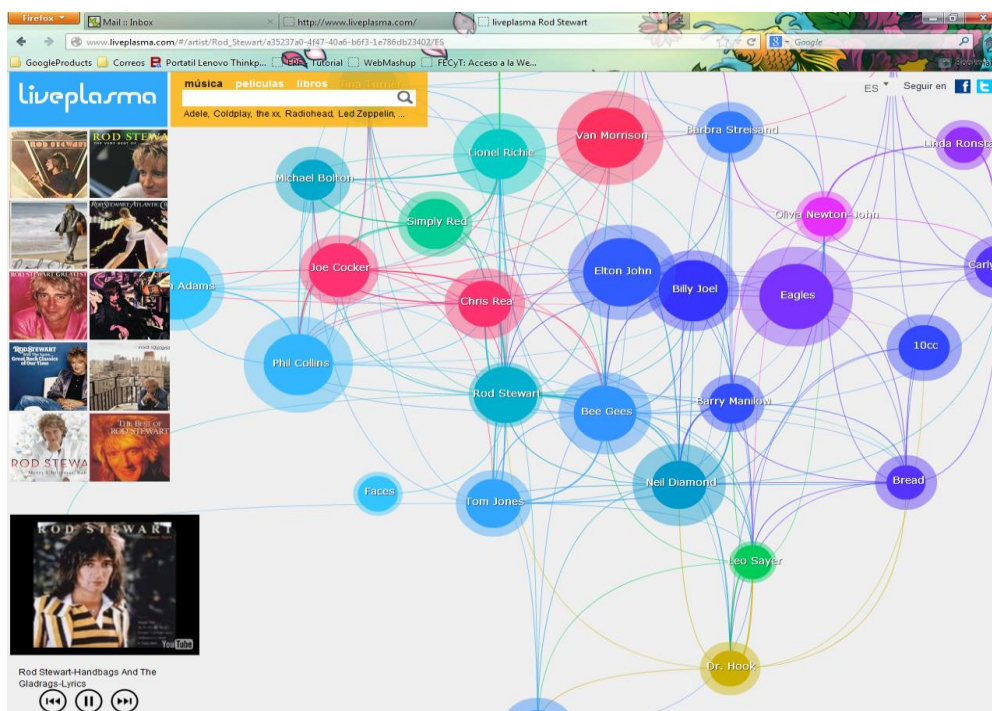


Figura 6-2 Pantalla principal Liveplasma

En la parte izquierda de la página encontramos debajo del logotipo de Liveplasma fotografías del artista provistas por una API de Amazon (Amazon Product Advertising), las mismas que cuando son seleccionadas mediante un clic, abren una página del navegador con toda la información para la compra del álbum de la fotografía, a través de Amazon.

En la parte inferior izquierda se encuentra un componente de Youtube, el mismo que permite reproducir el video de la canción, a la derecha se encuentra un grafo el mismo que vincula al actual artista con otros de su mismo género, que podrían interesar al usuario.

6.1.2. Caribbean Resorts Mashups

Caribbean Resorts Mashups es un Mashup bajo el dominio <http://www.gooseflap.net/>, que mezcla GoogleMaps con fotos de Panoramio para crear un nuevo modo de descubrir todos los puntos vacacionales de verano en el Caribe y México, incluye adicionalmente lugares vacacionales de Cuba y República Dominicana. Este brinda en un solo sitio detalles del lugar, acceso a comentarios, vistas aéreas, fotos de Panoramio y videos de YouTube.

Utiliza APIS de Google Maps y Panoramio y presenta la siguiente interface de usuario:

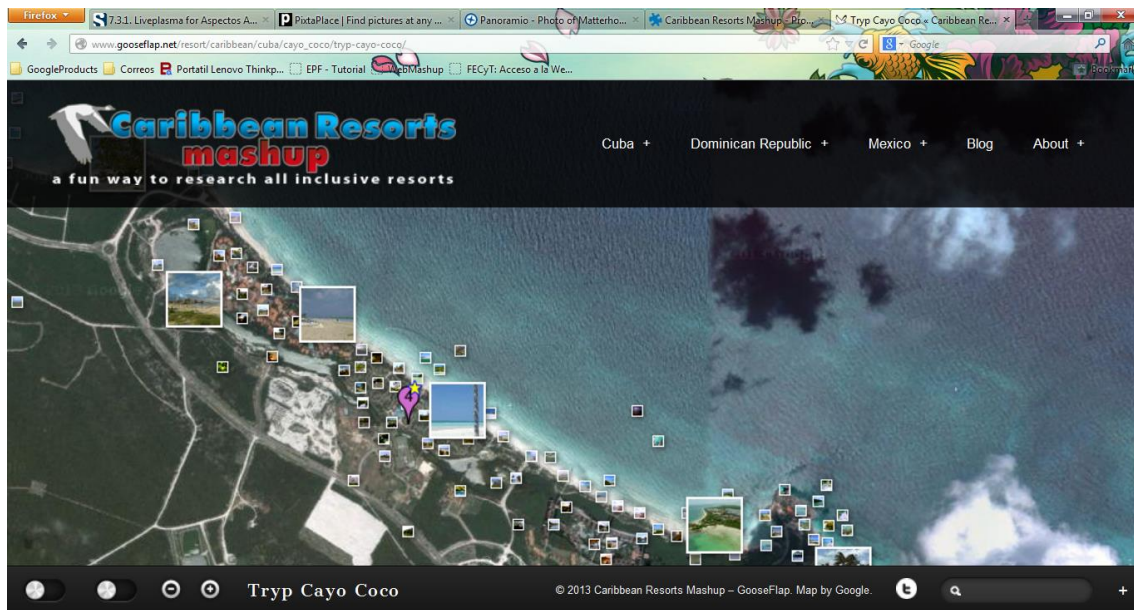


Figura 6-3 *Mashup Caribbean Resorts* <http://www.gooseflap.net/>

En los lugares en los que se aprecian cuadrados sobre el mapa encontramos fotos tomadas de Panoramio y encontramos también información adicional del lugar, revisiones y vínculos a Trip Advisor Page así como galerías más extensas de fotos, también encontramos APIs que vinculan el lugar con Facebook para dar “me gusta”, Tweets, G+.

Otra funcionalidad que ofrece es que se puede realizar selecciones de acuerdo al número de estrellas (3, 3.5, 4, 4.5, 5), a la marca de hoteles (Barcelo, Blue Diamond, Iberostar, Melia, Occidental, Palace), el tipo de entretenimiento que se puede encontrar (familiar, adulto solamente), al tamaño del lugar (grade, mediano o pequeño), al tipo de entretenimientos del sitio (casino, campo de golf, spa, etc.).

En la parte inferior se ubica un pequeño rectángulo con una foto del lugar, en el cual se pasa de ubicación en ubicación para poder revisar sitio por sitio y poder ver los detalles del mismo.

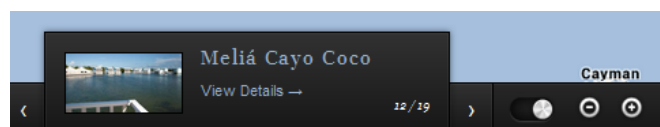


Figura 6-4 *Componente del Mashup Caribbean Resorts*

Cuando hacemos clic sobre “View Details”, podemos ver los detalles del sitio como se muestra en la siguiente imagen:

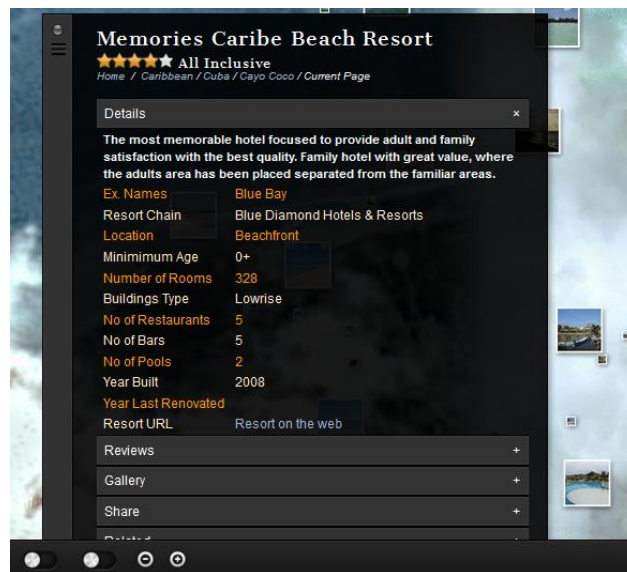


Figura 6-5 *Componente del Mashup Caribbean Resorts*

Dentro de Gallery podemos encontrar una galería de video, en la cual mediante el API de YouTube, pueden realizarse reproducciones del lugar en cuestión.

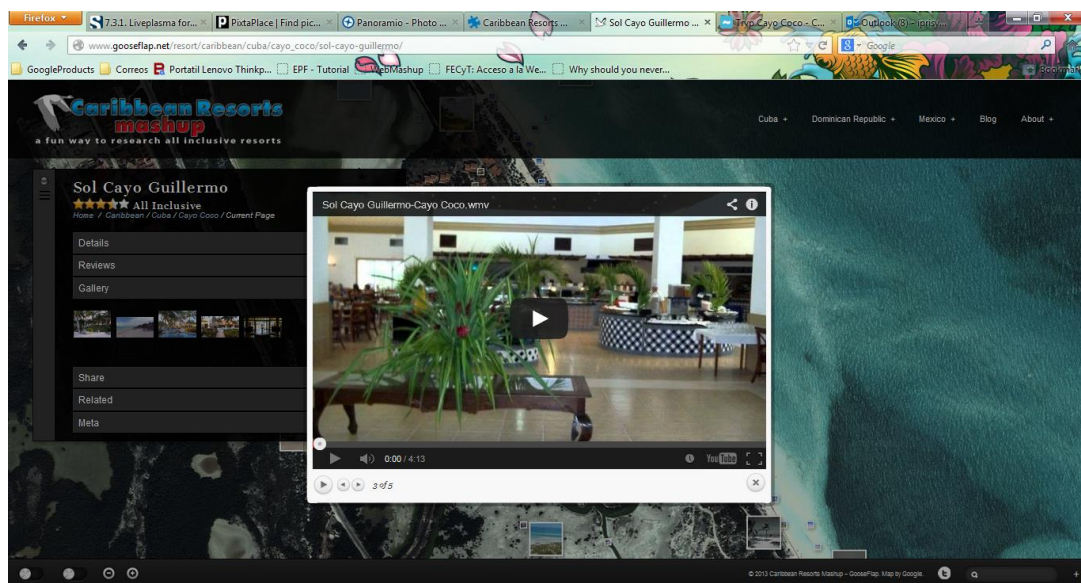


Figura 6-6 *Componente de YouTube del Mashup Caribbean Resorts Skypicker*

Bajo el dominio <http://www.skypicker.com/>, encontramos Spicker, este Mashup se trata de una máquina de búsqueda de tickets low-cost europea que muestra los resultados en Google Maps. La Herramienta tiene algunos filtros de búsqueda que hace que sea fácil encontrar el vuelo correcto.

La pantalla de bienvenida, muestra un pequeño diagrama con indicaciones del uso de la aplicación como se puede ver en la siguiente imagen:

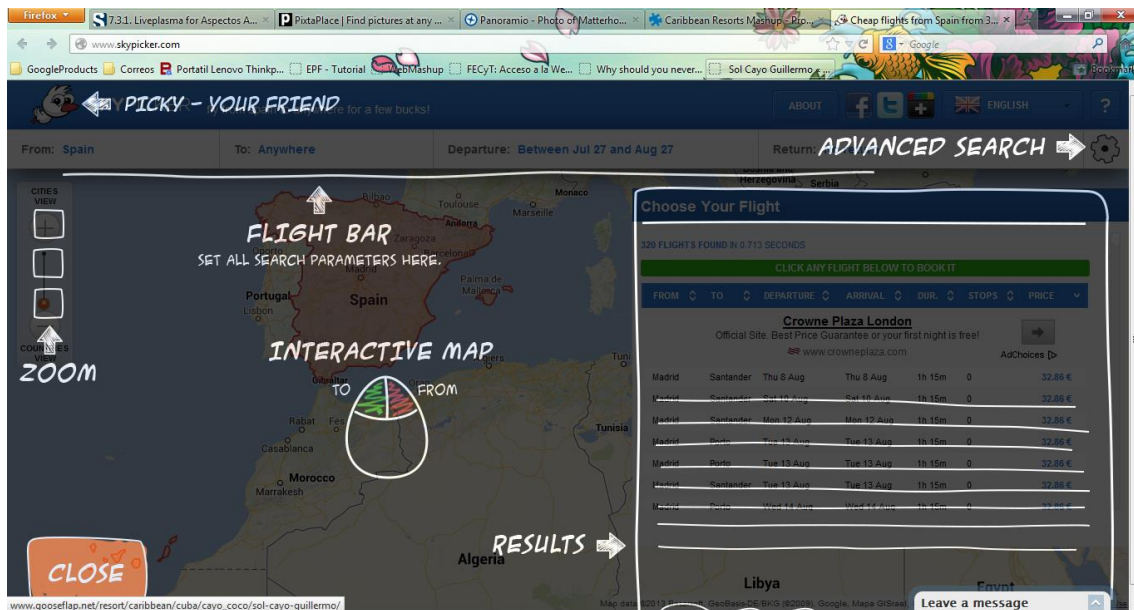


Figura 6-7 Mashup Skypicker www.skypicker.com

Una vez que se hace clic sobre la aplicación, la ayuda desaparece y en su lugar se visualiza el Mashup completo.

Encontramos en la esquina superior izquierda el logotipo de Skyper y a continuación información sobre la empresa desarrolladora del Mashup, luego vínculos hacia Facebook, Twitter y Google+, debemos tener en cuenta que son únicamente vínculos que llevan a dichas páginas, mas no consisten en APIs, una vez que se hace clic sobre uno de ellos abren el navegador en la página que representan.

A continuación se puede observar una lista desplegable de los idiomas en los cuales se desea ver la página.

A continuación, podemos seleccionar el lugar desde donde y hasta dónde queremos ir, la fecha de partida y la de retorno, para que según eso se listen las posibilidades en la matriz de la parte central en donde encontramos detalles del vuelo.



Figura 6-8 Parte superior Pantalla Skypicker

Se encuentra también un mapa en la parte central izquierda, con un control de zoom para la aplicación y a la derecha un listado de todos los vuelos que se pueden encontrar, en el cual se detalla el lugar de partida, de destino, la fecha de despegue y aterrizaje, la duración, el número de escalas y el precio.

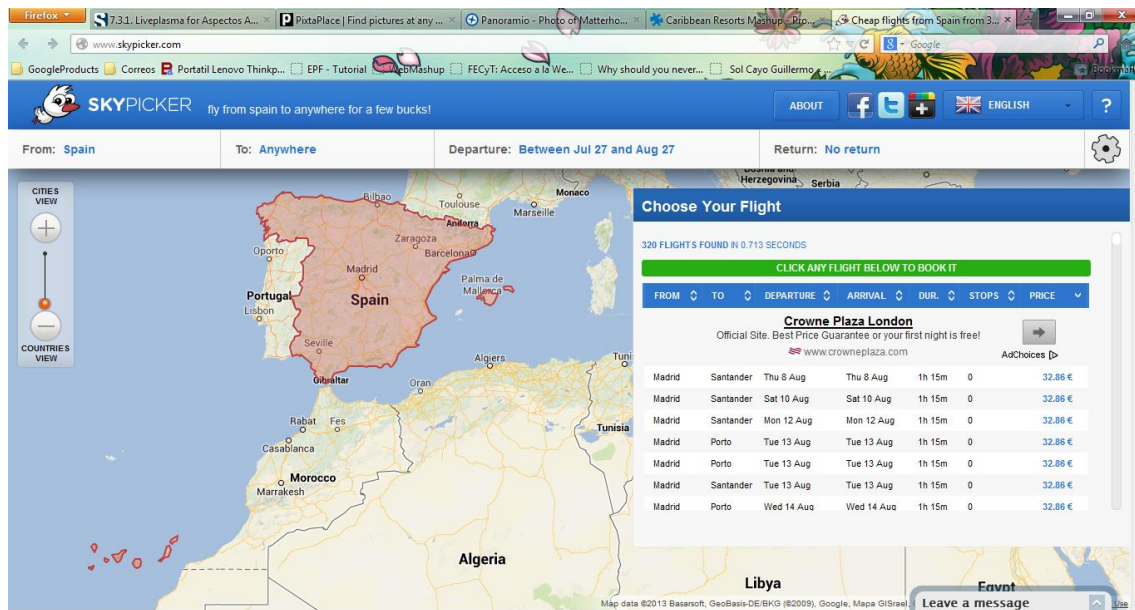


Figura 6-9 Pantalla Skypicker vista de componentes principales

En la parte inferior encontramos una opción “Leave a message”, la misma que permite dejar un mensaje a los operadores.

The screenshot shows a 'Leave a message' form. The form has a title 'Leave a message' and a sub-header 'Our operators are not available at the moment. Please leave a message and we will contact you soon'. Below this, there are three input fields: 'Your name:', 'E-mail:', and 'Message:'. Each field has a red asterisk indicating it is required. The 'Message' field is a larger text area. At the bottom of the form, there is a 'Leave a message' button. The form is styled with a light blue background and a white border.

Figura 6-10 Pantalla “leave a message” Skypicker

Cuando hacemos clic sobre una de las opciones de vuelo que nos interesan, aparecen detalles de la misma, en la que se presenta el itinerario del vuelo junto con el precio, podemos seleccionar sobre reservar el ticket y nos saldrá una pantalla con toda la información necesaria para poder ejecutar dicha acción e información adicional sobre equipaje, total del viaje, etc. También encontramos el modo de pago, si existen otros pasajeros, etc.

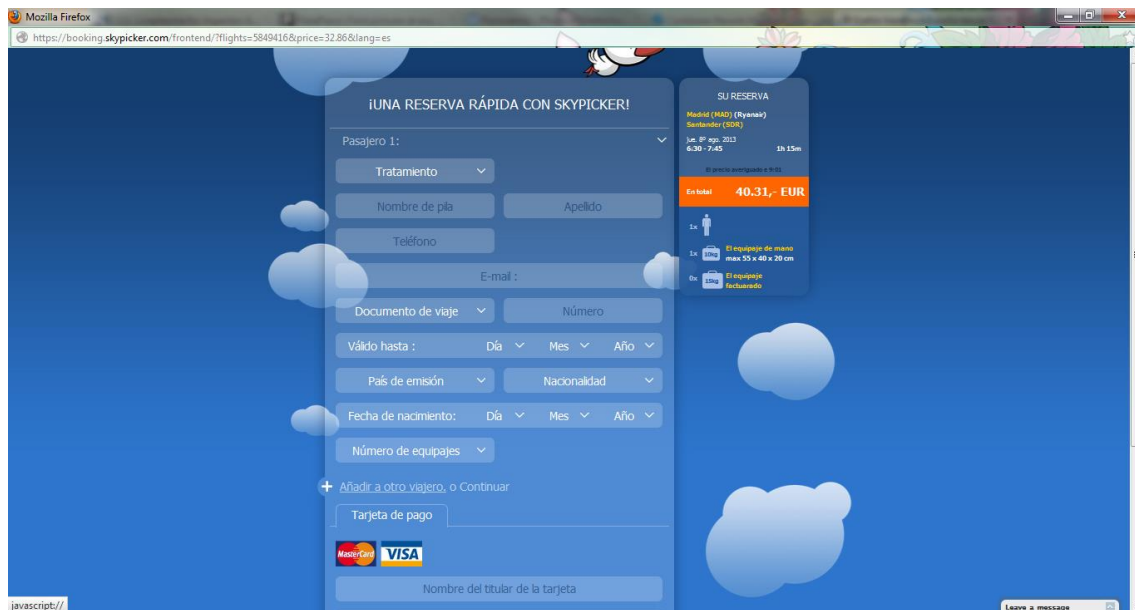


Figura 6-11 Pantalla de pago Skypicker

6.2. Aplicación del Método de Evaluación de Usabilidad

En esta sección se utilizará el método de evaluación propuesto en este trabajo en cada uno de los Mashups seleccionados. Se presentará paso a paso cada una de las fases y sus tareas involucradas al momento de la Evaluación, siguiendo la estructura presentada a continuación:

1. Requisitos de la evaluación.
2. Especificación de la Evaluación.
3. Diseño de la evaluación.
4. Ejecución de la evaluación e informe de usabilidad

6.2.1. Evaluación Liveplasma

En este apartado se evaluará la usabilidad del Mashup Liveplasma, teniéndose en cuenta los atributos relacionados directamente con el carácter compositivo del Mashup.

A continuación y con el fin de mejorar la explicación de la aplicación se listan los componentes del Mashup en cuestión:

5. Componente correspondiente al API de Amazon con la foto de los álbumes.
6. Componente de YouTube en donde se reproduce el video actual.
7. Componente para el motor de búsqueda de música, películas o libros.
8. Componente con el grafo de relaciones entre artistas.

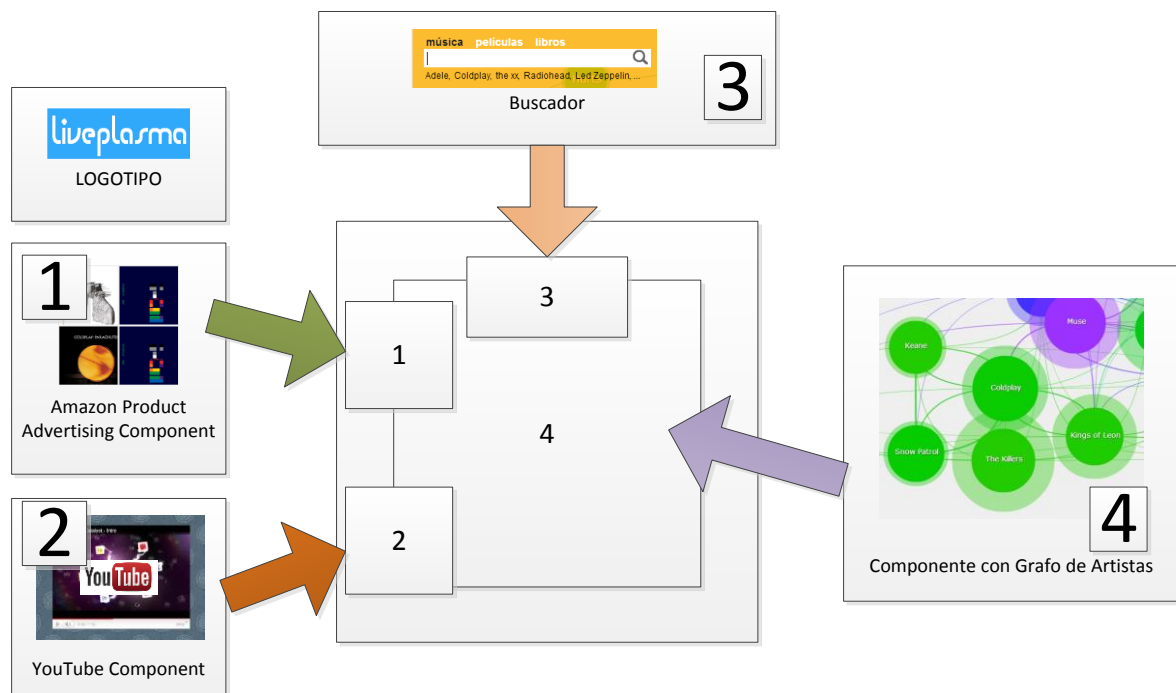


Figura 6-12 Disposición de componentes en Liveplasma

Fase 1: Especificación de los Requisitos de la Evaluación

En esta fase se establece el propósito de la evaluación; los perfiles relacionados con el Mashup, el estado del desarrollo en el que se enfocará la evaluación, el contexto de uso y la selección de los atributos de usabilidad. Todos estos aspectos vienen reflejados en el documento Requisitos de evaluación.

a) Establecimiento del Propósito de la Evaluación de la Usabilidad del Mashup

El objetivo es realizar una evaluación de la usabilidad sumativa de la aplicación en cuestión, es decir, se evaluará la aplicación que ya ha sido construida anteriormente, esta decisión se justifica ya que el Mashup está desarrollado con anterioridad y está disponible en la Web como una herramienta terminada.

b) Especificación de los Requisitos de Usabilidad del Mashup

Los requisitos de calidad que se requiere que cumpla la aplicación son los esperados a modo de ejemplo tanto por el desarrollador como por el usuario final, siendo estos dos los principales stakeholders envueltos en el método de evaluación descrito aquí.

Adicionalmente se seleccionarán los atributos de usabilidad específicos para Mashups, ya que se quiere remarcar especialmente este tipo de aplicaciones, para ello tendremos en cuenta cada uno de los atributos de usabilidad comentados y descritos en el Modelo de Usabilidad para Mashups y directamente vinculados con el carácter composicional de estas aplicaciones.

Por tanto los atributos a medirse son los siguientes:

Atributos a Evaluar	
1	Disposición de los componentes (1.1.4)
2	Tamaño del componente (1.1.5)
3	Coherencia en la agrupación de componentes (1.2.1)
4	Densidad de la información (1.2.2)
5	Popularidad de los componentes (1.3.4)
6	Presentación de controles en los componentes (1.3.5)
7	Actualización de información entre componentes (1.4.4)
8	Propósito fácilmente distinguible de los componentes (2.1.4)
9	Distinción de posibles acciones de los componentes (2.2.3)
10	Facilidad para alcanzar los componentes (3.2.8)
11	Carga de los componentes (3.2.9)
12	Comportamiento correcto de los componentes (3.4.6)
13	Independencia de dispositivos (3.5.1)
14	Auto-ajuste de los componentes en los dispositivos (3.5.2)

Tabla 6-1 **Atributos a evaluar en el Mashup Liveplasma**

c) Elaboración del Documento de Requisitos de Evaluación

En esta tarea fijaremos un documento que refleje los requisitos de la evaluación, para ello elaboraremos una plantilla, que contendrá la información de las tareas que se realizarán a los largo del método de evaluación.

En base a la Plantilla 1 del Anexo C “Documento de Especificación de Requisitos de Evaluación, se ha conseguido esta tarea según se muestra en la Plantilla 2 “Plantilla de Requisitos de Evaluación Liveplasma”

Fase 2: Especificación de la Evaluación

En esta fase seleccionaremos aquellos artefactos del Mashup que se desean evaluar y las métricas que serán empleadas. Las métricas seleccionadas van a ser calculadas y se establecerán umbrales para detectar los problemas de usabilidad. Luego definiremos la plantilla del informe para presentar los problemas de usabilidad encontrados. Todos estos aspectos se reflejan en el documento Especificación de la evaluación.

a) Selección de Artefactos a Evaluar

En el caso específico de este caso de estudio, evaluaremos el Mashup final, pero teniendo en cuenta los componentes que se encuentran en el mismo y la forma en que estos han sido unidos para formar la aplicación.

b) Selección de Métricas a Emplear y Definición del Criterio de Decisión para las Métricas de Usabilidad

Atendiendo a la selección de atributos de usabilidad y artefactos a evaluar, se han seleccionado las métricas que serán empleadas en este ejemplo y se presentan en la tabla 6-2.

La prioridad de resolución del problema de usabilidad se evaluará de acuerdo al valor obtenido tras el cálculo de la métrica.

Atributo	Métrica	Umbrales
1.1.4. Disposición de componentes (DC)	$\frac{\text{Nro. componentes totalmente visibles}}{\text{Nro. total de componentes a mostrarse}}$	Problema mayor de usabilidad: $0 \leq \text{valor} < 0.35$ Problema medio de usabilidad: $0.35 \leq \text{valor} < 0.75$ Problema menor de usabilidad: $0.75 \leq \text{valor} < 1$ Sin ningún problema de usabilidad: 1
1.1.5. Tamaño del componente (TC)	$\frac{\text{Nro. componentes con tamaño apropiado}}{\text{Nro. total de componentes}}$	Problema mayor de usabilidad: $0 \leq \text{valor} < 0.35$ Problema medio de usabilidad: $0.35 \leq \text{valor} < 0.75$ Problema menor de usabilidad: $0.75 \leq \text{valor} < 1$ Sin ningún problema de usabilidad: 1
1.2.1. Coherencia en la agrupación de los componentes (CAC)	$\frac{\text{Nro. componentes coherentes}}{\text{Nro. total de componentes}}$	Problema mayor de usabilidad: $0 \leq \text{valor} < 0.35$ Problema medio de usabilidad: $0.35 \leq \text{valor} < 0.75$ Problema menor de usabilidad: $0.75 \leq \text{valor} < 1$ Sin ningún problema de usabilidad: 1
1.2.2. Densidad de la información (DI)	Escala de Likert: La información se muestra en una forma adecuada y sin sobrecargar la página. 1. Totalmente de acuerdo. 2. Ni en acuerdo, ni en desacuerdo. 3. Totalmente en desacuerdo.	En este caso si en la evaluación se muestra la opción 3, se presenta un problema de usabilidad; en el caso de la opción 2, se presenta un problema menor de usabilidad; y por último en primer caso no existe problema ninguno de usabilidad.
1.3.4. Popularidad de los componentes (PC)	$\frac{\text{Nro. de componentes bien conocidos}}{\text{Nro. total de componentes}}$	Problema medio de usabilidad: $0 \leq \text{valor} < 0.5$ Problema menor de usabilidad: $0.5 \leq \text{valor} < 1$ Sin ningún problema de usabilidad: 1
1.3.5. Presentación de controles en componentes (PCC)	$\frac{\text{Controles usuales}}{\text{Nro. componentes que presentan controles}}$	Problema medio de usabilidad: $0 \leq \text{valor} < 0.7$ Problema menor de usabilidad: $0.7 \leq \text{valor} < 1$ Sin ningún problema de usabilidad: 1
1.4.4. Actualización de información entre componentes (AIC)	$\frac{\text{Componentes auto_sincronizados}}{\text{Nro. total comp. que deben autosincronizarse}}$	Problema mayor de usabilidad: $0 \leq \text{valor} < 0.35$ Problema medio de usabilidad: $0.35 \leq \text{valor} < 0.75$ Problema menor de usabilidad: $0.75 \leq \text{valor} < 1$ Sin ningún problema de usabilidad: 1
2.1.4. Propósito fácilmente distinguible de componentes (PFD)	$\frac{\text{Nro. comp. fácilmente distinguibles}}{\text{Nro. total de componentes}}$	Problema mayor de usabilidad: $0 \leq \text{valor} < 0.35$ Problema medio de usabilidad: $0.35 \leq \text{valor} < 0.75$ Problema menor de usabilidad: $0.75 \leq \text{valor} < 1$ Sin ningún problema de usabilidad: 1
3.2.8. Facilidad para alcanzar los componentes (FAC)	$\frac{\text{Componentes alcanzables}}{\text{Nro. total componentes}}$	Problema mayor de usabilidad: $0 \leq \text{valor} < 0.35$ Problema medio de usabilidad: $0.35 \leq \text{valor} < 0.75$ Problema menor de usabilidad: $0.75 \leq \text{valor} < 1$ Sin ningún problema de usabilidad: 1
3.2.9. Carga de los componentes (CC)	$\frac{\text{Nro. componentes sin dificultad de carga}}{\text{Nro. total componentes}}$	Problema mayor de usabilidad: $0 \leq \text{valor} < 0.35$ Problema medio de usabilidad: $0.35 \leq \text{valor} < 0.75$ Problema menor de usabilidad: $0.75 \leq \text{valor} < 1$ Sin ningún problema de usabilidad: 1
3.4.6. Comportamiento correcto de los componentes (CCC)	$\frac{\text{Nro. comp. comportamiento apropiado}}{\text{Nro. total componentes}}$	Problema mayor de usabilidad: $0 \leq \text{valor} \leq 0.5$ Problema medio de usabilidad: $0.5 < \text{valor} \leq 0.9$ Problema menor de usabilidad: $0.9 < \text{valor} < 1$ No existe un problema de usabilidad 1
3.5.1. Independencia de dispositivos (ID)	$\frac{\text{Nro. plataf. que conservan operatividad}}{\text{Nro. total de plataformas esperadas}}$	Problema mayor de usabilidad: $0 \leq \text{valor} < 0.35$ Problema medio de usabilidad: $0.35 \leq \text{valor} < 0.75$ Problema menor de usabilidad: $0.75 \leq \text{valor} < 1$ Sin ningún problema de usabilidad: 1
3.5.2. Auto-ajuste de los componentes en los dispositivos (ACD)	$\frac{\text{Resoluc. con auto - ajuste}}{\text{Nro resoluc. esperadas}}$	Problema mayor de usabilidad: $0 \leq \text{valor} < 0.35$ Problema medio de usabilidad: $0.35 \leq \text{valor} < 0.75$ Problema menor de usabilidad: $0.75 \leq \text{valor} < 1$ Sin ningún problema de usabilidad: 1

Tabla 6-2 Métricas a ser evaluadas en el caso de estudio Liveplasma

c) Definición del Criterio de Normalización y Agregación

En el caso de este caso de estudio, no definiremos aquí los criterios de normalización y agregación, puesto que estamos evaluando un Mashup en su fase de uso mas no de composición, por tanto no haría falta esta tarea.

d) Definición de la Plantilla para el Informe de Usabilidad

Para la presentación de problemas de usabilidad se usa una plantilla propuesta en el capítulo que especifica el método de evaluación.

Fase 3: Diseño de la Evaluación

En esta fase se indican todas aquellas restricciones de índole técnica que condicionan la evaluación, procediéndose a elaborar el plan de evaluación.

a) Definición de Restricciones

Las principales restricciones con las que se cuenta al momento de realizar la evaluación de este Mashup están relacionadas al hecho de estar evaluando un Mashup que al no ser elaborado directamente por el interesado en realizar la evaluación, no se cuenta con el código ni con la forma de desarrollar directa, para poder luego aplicar las medidas correctivas oportunas, otra restricción también es que se está realizando una evaluación en fase final, no pudiendo observarse el proceso de evaluación en el estado de composición mismo del Mashup, lo que sería de utilidad con el fin ilustrativo de este trabajo, sin embargo se ha tratado de alcanzar un nivel de detalle máximo a fin de cubrir todos los atributos posibles del modelo de usabilidad de tal modo que el lector pueda tener una idea clara de cómo utilizar las métricas y cómo realizar el proceso de evaluación en cualquier estado del Mashup.

b) Plan de evaluación

El plan de evaluación que se seguirá, será el de aplicar el Modelo de Usabilidad de Mashups, para ir evaluando el caso de estudio propuesto desde una visión global así como también si hace falta, aislando los componentes, es decir “fragmentando” en artefactos de software consistentes en cada componente como una caja negra para evaluar lo que se necesite desde la perspectiva individual de cada componente y cómo este influye en el producto final.

Fase 4: Ejecución de la Evaluación e Informe de Usabilidad

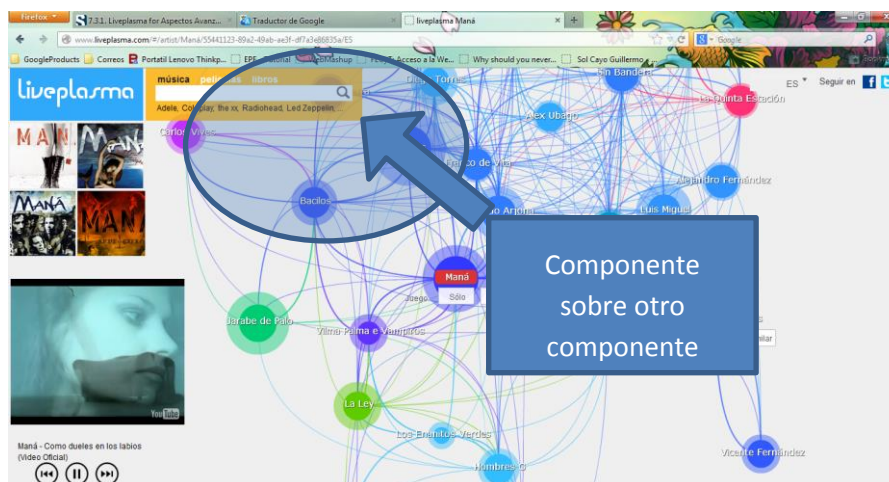
En esta fase se recibe la especificación detallada del plan de evaluación de la calidad así como los requisitos de calidad de la evaluación que se tienen ya disponibles, así también ya se han especificado las métricas para evaluar los artefactos y de pueden obtener los resultados para rellenar los informes de usabilidad correspondientes. Las tareas de esta fase son:

a) Evaluación de los Artefactos, Análisis de Cambios y Elaboración del Informe de Usabilidad

Una vez completados los pasos anteriores, se procede a realizar la medición de cada uno de los atributos escogidos para este caso de estudio. Como se puede observar en la Plantilla 2 en el cual se ve el documento de Especificación de los Requisitos de Evaluación de Liveplasma, en el que se plantea la medición en 13 atributos de calidad para este caso de estudio. Ellos se describen a continuación:

1. Disposición de los Componentes (1.1.4)

En cuanto a la disposición de los componentes en el presente caso de estudio, vemos que en la parte izquierda del Mashup se sitúan dos componentes y en el centro se observa el grafo con relaciones entre artistas, sin embargo se ve que el componente central se encuentra en algunas de sus secciones cubierto por otros componentes como es el caso del buscador en la parte superior, así podemos verlo en la figura a continuación:



Como se había explicado en el capítulo del Modelo de calidad, podemos calcular este atributo a través de la métrica:

$$DC = \frac{\text{Número de componentes totalmente visibles}}{\text{Número total de componentes a mostrarse}}$$

En este caso el número de componentes totalmente visibles son 3 y el grafo de relaciones entre artistas se ve solapado por los otros componentes, por lo cual no lo contamos como “componente totalmente visible”, aplicando la métrica tenemos:

$$DC = \frac{3}{4} = 0.75$$

La interpretación de la métrica es $0 \leq \text{valor} \leq 1$ en donde el valor más cercano a 1 es el mejor.

De acuerdo al grado de impacto de este atributo en la usabilidad del Mashup se establecen los siguientes umbrales:

Problema mayor de usabilidad: $0 \leq \text{valor} < 0.35$

Problema medio de usabilidad: $0.35 \leq \text{valor} < 0.75$

Problema menor de usabilidad: $0.75 \leq \text{valor} < 1$

Sin ningún problema de usabilidad: 1

En el caso de Liveplasma encontramos en cuanto a este atributo un problema menor de usabilidad.

ID	001
Descripción	El componente central del Mashup, en sus extremos está detrás de los demás componentes, lo que hace que muchas veces no pueda ser alcanzado en su totalidad.
Atributo Afectado	1. Facilidad de entendimiento / 1.1. Legibilidad visual / 1.1.4. Disposición de componentes.
Nivel de Criticidad	Menor.
Artefacto(s) evaluado(s)	Componente de grafo de artistas en conjunto con el Mashup final.
Ocurrencias	3 (una por cada componente que sobrepone al componente central)
Recomendaciones	Delimitar la zona en la que se muestra el componente central.
Prioridad	Baja.

Tabla 6-3 Problema de usabilidad 001 detectado.

2. Tamaño del Componente (1.1.5)

Como se estableció en el modelo de usabilidad este atributo ayuda a que los componentes sean útiles y usables de acuerdo a su tamaño ya que de esta manera pueden ser fácilmente manipulados y pueden mostrar la información necesaria sin esfuerzo por parte del usuario. La métrica para medir este atributo es:

$$TC = \frac{\text{Número de componentes con tamaño apropiado}}{\text{Número total de componentes}} = \frac{3}{4} = 0.75$$

De acuerdo al grado de impacto de este atributo en la usabilidad del Mashup se asignaron en el modelo de calidad los umbrales recomendados, los mismos que fueron mantenidos para este caso de estudio.

Problema mayor de usabilidad: $0 \leq \text{valor} < 0.35$

Problema medio de usabilidad: $0.35 \leq \text{valor} < 0.75$

Problema menor de usabilidad: $0.75 \leq \text{valor} < 1$

Sin ningún problema de usabilidad: 1

En el Mashup Liveplasma se encuentran los 4 componentes antes mencionados de los cuales los 3 tienen el tamaño apropiado, por lo cual al aplicar la métrica obtenemos el valor de 0.75, mostrándose un problema de usabilidad menor.

ID	002
Descripción	El componente central del Mashup es muy grande.
Atributo Afectado	1. Facilidad de entendimiento / 1.1. Legibilidad visual / 1.1.3. Tamaño del componente.
Nivel de Criticidad	Menor.
Artefacto(s) evaluado(s)	Componente de grafo de artistas en conjunto con el Mashup final.
Ocurrencias	1
Recomendaciones	Delimitar el tamaño del componente central.
Prioridad	Baja.

Tabla 6-4 Problema de usabilidad 002 detectado.

3. Coherencia en la Agrupación de Componentes (1.2.1)

En cuanto a este atributo podemos decir que los componentes se encuentran todos relacionados a la esencia del Mashup, aportando cada uno de ellos al propósito del mismo coherentemente. Al aplicar la fórmula como se puede observar a continuación el valor obtenido es 1, que de acuerdo a los umbrales de referencia establecidos en el Capítulo del Modelo de Usabilidad, no presenta ningún problema.

$$CAC = \frac{\text{Número de componentes coherente}}{\text{Número total de componentes}} = \frac{4}{4} = 1$$

4. Densidad de la Información (1.2.2)

Con respecto a la densidad de la información, la misma que tiene que ver con la cantidad de información mostrada al mismo tiempo, según el método de medición referencial, establecimos una escala de Likert en donde realizamos la siguiente pregunta:

La información se muestra en una forma adecuada y sin sobrecargar la página.

1. Totalmente de acuerdo.
2. Ni en acuerdo, ni en desacuerdo.
3. Totalmente en desacuerdo.

En este caso se escogió la opción 1, estableciéndose que no existe un problema de usabilidad.

5. Popularidad de los Componentes (1.3.4)

En el caso de la popularidad de los componentes en la cual como se explicó anteriormente, se comprueba si los componentes son conocidos, aceptados y comúnmente usados en la comunidad, para el caso de Liveplasma tenemos que tres de los cuatro componentes son usados comúnmente, el componente de Amazon, el de YouTube y el formato de búsquedas, por lo cual medimos de acuerdo a la métrica:

$$PC = \frac{\text{Número de componentes bien conocidos}}{\text{Número total de componentes}} = \frac{3}{4} = 0.75$$

Para este atributo, debemos tener en cuenta que no necesariamente el hecho de que el componente no sea muy conocido pueda afectar a la usabilidad, ya que un componente no tan conocido puede ser muy fácil e intuitivo de usar, sin embargo el usar componentes bien conocidos puede aportar al éxito del Mashup, ya que son componentes aceptados por la comunidad de usuario que no necesitan de un aprendizaje inicial y que ayudan a darle al Mashup una aceptación más rápida. Por esta razón los rangos establecidos son diferentes a los que hemos venido utilizando en atributos anteriores, teniendo:

Problema medio de usabilidad:	$0 \leq \text{valor} < 0.5$
Problema menor de usabilidad:	$0.5 \leq \text{valor} < 1$
Sin ningún problema de usabilidad:	1

Por tanto en Liveplasma y de acuerdo a este atributo observamos un problema menor de usabilidad al tener el 0.75 como resultado del cálculo.

ID	003
Descripción	El componente central del Mashup no es popular.
Atributo Afectado	1. Facilidad de entendimiento / 1.3. Familiaridad / 1.3.4. Popularidad de los componentes.
Nivel de Criticidad	Menor.
Artefacto(s) evaluado(s)	Componente de grafo de artistas en conjunto con el Mashup final.
Ocurrencias	1
Recomendaciones	En este caso se puede recomendar la búsqueda de algún componente para reproducción más popular, aunque no sea tan vistoso como lo es el actual.
Prioridad	Baja.

Tabla 6-5 Problema de usabilidad 003 detectado

6. Presentación de Controles en Componentes (1.3.5)

Como se había indicado en el Capítulo 4, en este atributo se toma en cuenta si los controles en los componentes son usuales, midiendo este atributo a través de la relación existente entre los componentes con controles usuales y el número total de componentes que presentan controles.

$$PCC = \frac{\text{Controles usuales}}{\text{Nro.Total de Componentes que presentan controles}} = \frac{2}{3} = 0.67$$

En el caso de Liveplasma, encontramos dos componentes que presentan controles conocidos y aceptados: en primer lugar el buscador, que presenta el típico control de lupa para iniciar una búsqueda y luego la API de YouTube que tiene consigo los botones para iniciar, pausar, adelantar o retroceder la reproducción musical.

Los umbrales elegidos en este caso son:

Problema medio de usabilidad: $0 \leq \text{valor} < 0.7$
 Problema menor de usabilidad: $0.7 \leq \text{valor} < 1$
 Sin ningún problema de usabilidad: 1

ID	004
Descripción	El componente central del Mashup no presenta controles usuales.
Atributo Afectado	1. Facilidad de entendimiento / 1.3. Familiaridad / 1.3.5. Presentación de controles en componentes.
Nivel de Criticidad	Menor.
Artefacto(s) evaluado(s)	Componente de grafo de artistas en conjunto con el Mashup final.
Ocurrencias	2
Recomendaciones	Existen dos botones que son relacionados a la reproducción del artista que está en el círculo y otro botón para reproducir música de artistas de similar género, la recomendación sería buscar mejores controles, más intuitivos y más usuales.
Prioridad	Baja.

Tabla 6-6 Problema de usabilidad 004 detectado.

Por tanto en Liveplasma y de acuerdo a este atributo observamos un problema menor de usabilidad al tener el 0.67 como resultado del cálculo, ya que el componente de grafo no presenta controles usuales, ya que sus botones son muy particulares al desarrollador del componente.

7. Actualización de Información entre Componentes (1.4.4)

Como se describió en el modelo de usabilidad este atributo evalúa si los componentes interconectados son actualizados automáticamente, esto es, en el caso de que se refresque por una búsqueda o acción de usuario que requiera dicha actualización en todos los componentes del Mashup, esta actualización se mide de acuerdo a la métrica planteada y en el caso de este ejemplo sería:

$$AIC = \frac{\text{Componentes auto_sincronizados}}{\text{Nro.total componentes que deben autosincronizarse}} = \frac{1}{3} = 0.33$$

De acuerdo al grado de impacto de este atributo en la usabilidad del Mashup se establecen los siguientes umbrales:

Problema mayor de usabilidad:	$0 \leq \text{valor} < 0.35$
Problema medio de usabilidad:	$0.35 \leq \text{valor} < 0.75$
Problema menor de usabilidad:	$0.75 \leq \text{valor} < 1$
Sin ningún problema de usabilidad:	1

En este caso, el problema de usabilidad es mayor, ya que el componente YouTube, no se sincroniza con los otros en el caso de realizarse una nueva búsqueda o elegirse un nuevo artista para ser reproducido y en el caso de elegir un botón del grafo, las imágenes no se actualizan correctamente.

ID	005
Descripción	El componente central del Mashup no presenta controles usuales.
Atributo Afectado	1. Facilidad de entendimiento / 1.4. Reducción de esfuerzo / 1.4.4. Actualización de información entre componentes.
Nivel de Criticidad	Mayor.
Artefacto(s) evaluado(s)	Componentes que forman parte del Mashup
Ocurrencias	3
Recomendaciones	Actualizar los componentes para que siempre concuerden entre sí, lo que se escucha y lo que se muestra debe concordar.
Prioridad	Alta.

Tabla 6-7 Problema de usabilidad 005 detectado

8. Propósito Fácilmente Distinguible de los Componentes (2.1.4)

Como se había visto en el modelo de usabilidad, este atributo trata sobre la facilidad de que el usuario entienda cual es el propósito del componente dentro del Mashup de una manera sencilla, la métrica que se utiliza es:

$$PFD = \frac{\text{Nro. componentes fácilmente distinguibles}}{\text{Nro. total de componentes}} = \frac{3}{4} = 0,75$$

En el caso del Mashup en cuestión, el número de componentes total es 4 y el número de componentes con un propósito fácilmente distinguible es 3, contándose dentro de los fácilmente distinguibles el componente de búsqueda, el de YouTube y por último el Grafo, no contamos como fácilmente distinguible el componente de Amazon, ya que no es muy claro el porqué de las fotografías a la izquierda, siendo muy poco intuitivo su propósito.

La valoración de la usabilidad la haremos utilizando los siguientes umbrales:

Problema mayor de usabilidad:	$0 \leq \text{valor} < 0.35$
Problema medio de usabilidad:	$0.35 \leq \text{valor} < 0.75$
Problema menor de usabilidad:	$0.75 \leq \text{valor} < 1$
Sin ningún problema de usabilidad:	1

Siendo para el caso de Liveplasma un problema menor de usabilidad el valor obtenido de este atributo luego de la aplicación de la métrica.

ID	006
Descripción	El componente de Amazon no muestra un propósito claro, que es el de vincularlo con la venta de los álbumes de los artistas en reproducción.
Atributo Afectado	2. Facilidad de aprendizaje / 2.1. Predictibilidad/ 2.1.4. Propósito fácilmente distinguible de los componentes.
Nivel de Criticidad	Menor.
Artefacto(s) evaluado(s)	Componentes que forman parte del Mashup, específicamente la API de Amazon.
Ocurrencias	1
Recomendaciones	Establecer el propósito del componente Amazon más claramente, para que sea más eficaz lo que se quiere conseguir.
Prioridad	Menor.

Tabla 6-8 Problema de usabilidad 006 detectado

9. Distinción de Posibles Acciones de los Componentes (2.2.3)

Con respecto a la distinción de posibles acciones de los componentes, vimos que este nos permite evaluar si los componentes muestran sus acciones de una forma claramente identificable dentro del Mashup. La métrica asociada es la relación entre los componentes cuyas acciones son fácilmente identificables entre el número total de componentes. La métrica asociada a este atributo es:

$$DPA = \frac{\text{Acciones de componentes fácilmente identificadas}}{\text{Nro. total de componentes}} = \frac{3}{4} = 0.75$$

En el caso de Liveplasma, se ha contado como tres el número de acciones de componentes fácilmente identificadas debido a que el componente de Amazon no hace que sea fácil distinguir que al hacer clic sobre una foto, este nos permite ir a Amazon para poder adquirir el álbum que está sonando actualmente. Por tanto el valor obtenido es 0.75, que según los umbrales fijados y mostrados a continuación, representa un problema menor de usabilidad.

La valoración de la usabilidad se realiza utilizando los siguientes umbrales:

Problema mayor de usabilidad: $0 \leq valor < 0.35$
 Problema medio de usabilidad: $0.35 \leq valor < 0.75$
 Problema menor de usabilidad: $0.75 \leq valor < 1$
 Sin ningún problema de usabilidad: 1

ID	007
Descripción	El componente de Amazon no muestra ninguna acción, parece solamente una galería de fotografías pero no se nota que detrás de ellas haya una acción determinada.
Atributo Afectado	2. Facilidad de aprendizaje / 2.2. Potencialidad/ 2.2.3. Distinción de posibles acciones de componentes.
Nivel de Criticidad	Menor.
Artefacto(s) evaluado(s)	Componentes que forman parte del Mashup, específicamente la API de Amazon.
Ocurrencias	1
Recomendaciones	Indicar de alguna manera que al hacer clic sobre la foto del álbum del artista, se tendrá la opción de compra del mismo.
Prioridad	Menor.

Tabla 6-9 Problema de usabilidad 007 detectado

10. Facilidad para Alcanzar los Componentes (3.2.8)

Este atributo mide si el Mashup es usable en cuanto a la facilidad de alcanzar todos los componentes en el Mashup, su métrica es la relación que existe entre todos los componentes que pueden ser alcanzados fácilmente y el número total de componentes.

$$FAC = \frac{\text{Componentes alcanzables}}{\text{Nro. total componentes}} = \frac{3}{4} = 0.75$$

Se ha puntuado a este Mashup con un valor de 3 entre 4 debido a que el componente del grafo central, muchas veces se extiende más allá de lo que debería, ocultándose parcialmente dentro de los otros controles, lo que hace que los vínculos escondidos no sean alcanzables por parte de los usuarios sin necesidad de acciones adicionales como arrastrar todo el grafo para poder ver las partes ocultas.

En el caso de este atributo tenemos los siguientes umbrales:

Problema mayor de usabilidad: $0 \leq valor < 0.35$
 Problema medio de usabilidad: $0.35 \leq valor < 0.75$
 Problema menor de usabilidad: $0.75 \leq valor < 1$
 Sin ningún problema de usabilidad: 1

Como se puede observar el valor resultante cae dentro del rango de *Problema menor de usabilidad*.

ID	008
Descripción	El grafo central se extiende más allá de los límites que deberían muchas veces siendo difícil llegar a todos los segmentos del grafo, puesto que estos están solapados por los otros componentes.
Atributo Afectado	3. Operatividad/ 3.2. Controlabilidad/ 2.2.8. Facilidad para alcanzar los componentes.

Nivel de Criticidad	Menor.
Artefacto(s) evaluado(s)	Componente de grafo del Mashup (Central)
Ocurrencias	1
Recomendaciones	Delimitar las zonas en las que se extiende el grafo.
Prioridad	Menor.

Tabla 6-10

Problema de usabilidad 008 detectado

11. Carga de los componentes (1.2.9)

La carga de componentes va relacionada con la dificultad en la carga de estos y se ha definido la métrica:

$$CC = \frac{\text{Nro. componentes sin dificultad de carga}}{\text{Nro. total componentes}} = \frac{4}{4} = 1$$

Para el caso de Liveplasma, hemos puntuado al número de componentes sin dificultad de carga con el valor de 4 cuando lo ejecutamos en un entorno Web ya que para entornos móviles este Mashup no está disponible.

Al ser los umbrales los siguientes:

Problema mayor de usabilidad: $0 \leq \text{valor} < 0.35$

Problema medio de usabilidad: $0.35 \leq \text{valor} < 0.75$

Problema menor de usabilidad: $0.75 \leq \text{valor} < 1$

Sin ningún problema de usabilidad: 1

No se ha encontrado ningún problema de usabilidad con respecto a este atributo.

12. Comportamiento Correcto de los Componentes (3.4.6)

En este atributo se evalúa si los componentes se comportan correctamente según sus entradas de datos / eventos. Para ello utilizamos la siguiente métrica:

$$CCC = \frac{\text{Nro. componentes comportamiento apropiado}}{\text{Nro. total componentes}} = \frac{1}{4} = 0.25$$

Al ser muy importante el comportamiento conjunto de los componentes, se establecieron los siguientes umbrales:

Problema mayor de usabilidad: $0 \leq \text{valor} \leq 0.5$

Problema medio de usabilidad: $0.5 < \text{valor} \leq 0.9$

Problema menor de usabilidad: $0.9 < \text{valor} < 1$

No existe un problema de usabilidad 1

En el caso de Liveplasma, los componentes presentan muchos fallos al momento de su sincronización y funcionamiento, siendo el caso por ejemplo que cuando se realiza una búsqueda de un nuevo artista y se introduce un dato erróneo o un artista inexistente, el componente de búsqueda deja de funcionar completamente,

cuando se elige un nuevo artista en el grafo, no se actualizan las fotografías, sino únicamente el video y por último cuando se presionan los botones del componente de YouTube para adelantar una reproducción con doble clic seguido el componente empieza a reproducir uno tras otro los videos sin haber forma de pararlo, todo esto cuando se ejecuta el Mashup en el navegador Firefox, se realizaron sin embargo pruebas del mismo en otros navegadores, concretamente en Google Chrome y no ocurrió este fallo.

En el caso de este Mashup, el problema de usabilidad según este atributo es mayor.

ID	009
Descripción	Los componentes tienen errores al momento de la sincronización: 1. Si la búsqueda no produce resultados el componente de búsqueda deja de funcionar completamente. 2. Cuando se elige un nuevo artista en el grafo, no se actualizan las fotografías, solamente el video. 3. Cuando se presionan los botones del componente de YouTube para adelantar o retroceder, el componente funciona mal, entra en un bucle infinito de adelanto o retroceso.
Atributo Afectado	3. Operatividad / 3.4. Comportamiento correcto de los componentes / 3.4.6. Comportamiento correcto de los componentes.
Nivel de Criticidad	Mayor.
Artefacto(s) evaluado(s)	Todos los componentes del Mashup.
Ocurrencias	4
Recomendaciones	1. Arreglar el control de búsqueda para que no se dañe cuando no encuentra resultados de las búsquedas. 2. Revisar la sincronización entre los componentes. 3. Revisar si existe alguna actualización del API de YouTube, o ver el por qué este no está funcionando de forma adecuada en Firefox e indicar al usuario este detalle. Este control se probó tanto en Firefox como en Google Chrome, sin embargo el componente presentó problemas en Firefox, en Google Chrome funcionó adecuadamente.
Prioridad	Mayor.

Tabla 6-11 Problema de usabilidad 009 detectado

13. Independencia de dispositivos (3.5.1)

Este atributo aborda el tema de que si la operatividad de los Mashups es la misma a través de diferentes plataformas, para ello se utiliza la métrica de la relación existente entre las plataformas en las que el Mashup conserva la misma operatividad, para el caso de Liveplasma se ha probado en tres plataformas: Windows, MAC OS para IPAD y ANDROID, en la primera el MASHUP tiene una operatividad adecuada; en la segunda, se realizaron pruebas sobre un IPAD 3, en el cual no se ejecutó de forma correcta debido a que Adobe Flash Player no pudo ser soportado sobre este dispositivo, y por último en la tercera plataforma, siendo el Mashup ejecutado en un móvil Samsung Nexus S I9023, con la versión de Android 4.1.1 y en un móvil Samsung Galaxy 3 GT-i9190 Mini con la versión de Android 4.1.2, el Mashup no se ha ejecutado, debido a que requiere una versión de ADOBE FLASH PLAYER no disponible para este sistema operativo.

La métrica para evaluar este atributo es:

$$ID = \frac{\text{Nro.plataformas que conservan operatividad}}{\text{Nro.total de plataformas en que se espera igual operatividad}} = \frac{1}{3} = 0.33$$

Los umbrales establecidos para esta métrica son:

Problema mayor de usabilidad: $0 \leq \text{valor} < 0.35$

Problema medio de usabilidad: $0.35 \leq \text{valor} < 0.75$

Problema menor de usabilidad: $0.75 \leq \text{valor} < 1$

Sin ningún problema de usabilidad: 1

Estableciéndose un valor de 0.33 que representa un problema mayor de usabilidad.

ID	010
Descripción	El Mashup se ejecuta únicamente en navegadores Web tradicionales con sistemas operativos que soporten Adobe Flash Player, entre ellos ningún dispositivo móvil pudo ejecutar este Mashup.
Atributo Afectado	3. Operatividad / 3.5. Ejecuciones Portables / 3.5.1. Independencia de dispositivos.
Nivel de Criticidad	Mayor
Artefacto(s) evaluado(s)	Mashup final.
Ocurrencias	3
Recomendaciones	Buscar un componente sustituto que permita haga que el Mashup sea compatible con más dispositivos.
Prioridad	Mayor

Tabla 6-12 Problema de usabilidad 010 detectado

14. Auto-ajuste de los Componentes en los Dispositivos (3.5.2)

Para evaluar este atributo se debe comprobar si los componentes cambian de tamaño de acuerdo a las diferentes plataformas, dispositivos o tamaño de pantalla. La métrica que puede ser empleada para este atributo es:

$$ACD = \frac{\text{Resoluciones en las que se auto – ajustan los componentes}}{\text{Resoluciones para las que se desea portabilidad}} = \frac{6}{6} = 1$$

Los umbrales establecidos para esta métrica son:

Problema mayor de usabilidad: $0 \leq \text{valor} < 0.35$

Problema medio de usabilidad: $0.35 \leq \text{valor} < 0.75$

Problema menor de usabilidad: $0.75 \leq \text{valor} < 1$

Sin ningún problema de usabilidad: 1

La métrica anterior trata de resoluciones, dependiendo de los dispositivos en los que se requiera ejecución y adicionalmente las resoluciones de dichos dispositivos se fijarán los valores para el denominador de esta métrica. En el caso de Liveplasma, se han contado como ejemplo las resoluciones de pantalla en las que se requiere que no cambie y que la

operatividad del Mashup no se vea afectada, por redistribución o cambio de tamaño de los componentes del Mashup.

Para el caso de este Mashup, se han tenido en cuenta resoluciones de ordenadores mas no las de otros dispositivos, ya que el Mashup no es portable a otras plataformas. Liveplasma no presenta problema de usabilidad en este atributo utilizándolo en las plataformas permitidas.

6.2.2. Evaluación del Mashup Caribbean Resorts

En este apartado se evaluará la usabilidad del Mashup Caribbean Resorts, teniéndose en cuenta los atributos relacionados directamente con el carácter composicional del Mashup.

A continuación y con el fin de mejorar la explicación de la aplicación se listan los componentes del Mashup en cuestión:

1. Componente de detalles.
2. API de Google Maps
3. API de Panoramio
4. Detalle de Viajes
5. API de YouTube

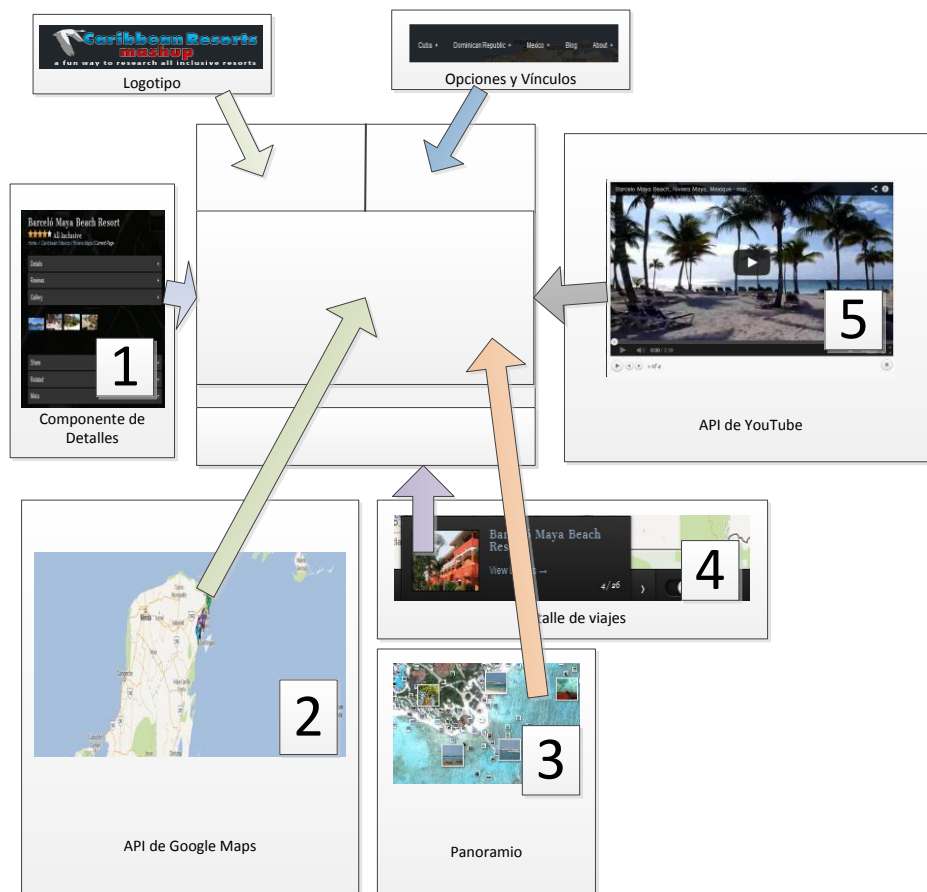


Figura 6-13

Lista de componentes Caribbean Resorts Mashup

Fase1: Especificación de los Requisitos de la Evaluación

En esta fase se establece el propósito de la evaluación; los perfiles relacionados con el Mashup, el estado del desarrollo en el que se enfocará la evaluación, el contexto de uso y la selección de los atributos de usabilidad. Todos estos aspectos vienen reflejados en el documento requisitos de evaluación.

a) Establecimiento del Propósito de la Evaluación

El objetivo es realizar una evaluación de la usabilidad sumativa de la aplicación en cuestión, es decir, se evaluará una aplicación que ya ha sido construida anteriormente, el Mashup está desarrollado con anterioridad y está disponible en la Web como una herramienta terminada.

b) Especificación de los Requisitos de Usabilidad del Mashup

Los requisitos de calidad que se requiere que cumpla la aplicación son los esperados a modo de ejemplo tanto por el desarrollador como por el usuario final, siendo estos dos los principales stakeholders envueltos en el método de evaluación descrito aquí.

Adicionalmente se seleccionarán los atributos de usabilidad específicos para Mashups, ya que se quiere remarcar especialmente este tipo de aplicaciones, para ello tendremos en cuenta ciertos atributos de usabilidad comentados y descritos en el Modelo de Usabilidad para Mashups y directamente vinculados con el carácter composicional de estas aplicaciones.

Los atributos a medirse son los siguientes:

Atributos a Evaluar	
1	Disposición de los componentes (1.1.4)
2	Tamaño del componente (1.1.5)
3	Coherencia en la agrupación de componentes (1.2.1)
4	Popularidad de los componentes (1.3.4)
5	Actualización de información entre componentes (1.4.4)
6	Propósito fácilmente distinguible de los componentes (2.1.4)

c) Elaboración del Documento de Requisitos de Evaluación

En esta tarea fijaremos un documento que refleje los requisitos de la evaluación, para ello elaboraremos una plantilla, que contendrá la información de las tareas que se realizarán a los largo del proceso de evaluación.

En base a la Plantilla 1, se ha realizado esta tarea según se muestra en la Plantilla 3 “Plantilla de Requisitos de Evaluación Caribbean Resorts”

Fase 2: Especificación de la Evaluación

En esta fase seleccionaremos aquellos artefactos del Mashup que se desean evaluar y las métricas que serán empleadas. Las métricas seleccionadas van a ser calculadas y se utilizarán los mismos umbrales definidos en el modelo de calidad para detectar los problemas de

usabilidad. Luego realizaremos la plantilla del informe para presentar los problemas de usabilidad encontrados. Todos estos aspectos se reflejan en el documento de especificación de la evaluación.

a) Selección de Artefactos a Evaluar

En el caso específico de este caso de estudio, evaluaremos el Mashup final, pero teniendo en cuenta los componentes que se encuentran en el mismo y la forma en que estos han sido unidos para formar la aplicación.

b) Métricas a Emplear y Definición del Criterio de Decisión para las Métricas de Usabilidad

Atendiendo a la selección de atributos de usabilidad y artefactos a evaluar, se han seleccionado las métricas que serán empleadas en este ejemplo y se presentan en la siguiente tabla:

Atributo	Métrica	Umbrales
1.1.4. Disposición de componentes (DC)	$\frac{\text{Nro. componentes totalmente visibles}}{\text{Nro. total de componentes a mostrarse}}$	Problema mayor de usabilidad: $0 \leq \text{valor} < 0.35$ Problema medio de usabilidad: $0.35 \leq \text{valor} < 0.75$ Problema menor de usabilidad: $0.75 \leq \text{valor} < 1$ Sin ningún problema de usabilidad: 1
1.1.5. Tamaño del componente (TC)	$\frac{\text{Nro. componentes con tamaño apropiado}}{\text{Nro. total de componentes}}$	Problema mayor de usabilidad: $0 \leq \text{valor} < 0.35$ Problema medio de usabilidad: $0.35 \leq \text{valor} < 0.75$ Problema menor de usabilidad: $0.75 \leq \text{valor} < 1$ Sin ningún problema de usabilidad: 1
1.2.1. Coherencia en la agrupación de los componentes (CAC)	$\frac{\text{Nro. componentes coherentes}}{\text{Nro. total de componentes}}$	Problema mayor de usabilidad: $0 \leq \text{valor} < 0.35$ Problema medio de usabilidad: $0.35 \leq \text{valor} < 0.75$ Problema menor de usabilidad: $0.75 \leq \text{valor} < 1$ Sin ningún problema de usabilidad: 1
1.3.4. Popularidad de los componentes (PC)	$\frac{\text{Nro. de componentes bien conocidos}}{\text{Nro. total de componentes}}$	Problema medio de usabilidad: $0 \leq \text{valor} < 0.5$ Problema menor de usabilidad: $0.5 \leq \text{valor} < 1$ Sin ningún problema de usabilidad: 1
1.4.4. Actualización de información entre componentes (AIC)	$\frac{\text{Componentes auto_sincronizados}}{\text{Nro. total comp. que deben autosincronizarse}}$	Problema mayor de usabilidad: $0 \leq \text{valor} < 0.35$ Problema medio de usabilidad: $0.35 \leq \text{valor} < 0.75$ Problema menor de usabilidad: $0.75 \leq \text{valor} < 1$ Sin ningún problema de usabilidad: 1
2.1.4. Propósito fácilmente distinguible de componentes (PFD)	$\frac{\text{Nro. comp. fácilmente distinguibles}}{\text{Nro. total de componentes}}$	Problema mayor de usabilidad: $0 \leq \text{valor} < 0.35$ Problema medio de usabilidad: $0.35 \leq \text{valor} < 0.75$ Problema menor de usabilidad: $0.75 \leq \text{valor} < 1$ Sin ningún problema de usabilidad: 1

Figura 6-14 Métricas a ser evaluadas en el caso de estudio Caribbean Resorts

La prioridad de resolución del problema de usabilidad se evaluará de acuerdo al valor obtenido tras el cálculo de la métrica.

c) Definición del Criterio de Normalización y Agregación

En el caso de este caso de estudio, no definiremos aquí los criterios de normalización y agregación, puesto que estamos evaluando un Mashup en su fase de uso mas no de composición, por tanto no haría falta esta tarea.

d) Definición de la Plantilla para el Informe de Usabilidad

Para la presentación de problemas de usabilidad se usa una plantilla propuesta en el capítulo que especifica el método de evaluación.

Fase 3: Diseño de la Evaluación

En esta fase se indican todas aquellas restricciones de índole técnica que condicionan la evaluación, procediéndose a elaborar el plan de evaluación.

a) Definición de Restricciones

Las principales restricciones al momento de realizar la evaluación de este Mashup están relacionadas al hecho de estar evaluando un Mashup que al no ser elaborado directamente por el interesado en realizar la evaluación, no se cuenta con el código ni con la forma de desarrollar directa, para poder luego aplicar las medidas correctivas oportunas, otra restricción también es que se está realizando una evaluación en fase final, no pudiendo observarse el proceso de evaluación en el estado de composición mismo del Mashup, lo que sería de utilidad con el fin ilustrativo de este trabajo.

b) Elaboración del Plan de Evaluación

El plan de evaluación que se seguirá, será el de aplicar el Modelo de Usabilidad de Mashups, para ir evaluando el caso de estudio propuesto desde una visión global así como también si hace falta, aislando los componentes, es decir “fragmentando” en artefactos de software consistentes en cada componente como una caja negra para evaluar lo que se necesite desde la perspectiva individual de cada componente y cómo este influye en el producto final.

Fase 4: Ejecución de la Evaluación e Informe de Usabilidad

En esta fase se recibe la especificación detallada del plan de evaluación de la calidad así como los requisitos de calidad de la evaluación que se tienen ya disponibles, así también ya se han especificado las métricas para evaluar los artefactos y de pueden obtener los resultados para rellenar los informes de usabilidad correspondientes. Las tareas de esta fase son:

a) Evaluación de los Artefactos, Análisis de Cambios y Elaboración del Informe de Usabilidad

Una vez completados los pasos anteriores, se procede a realizar la medición de cada uno de los atributos escogidos para este caso de estudio. Como se puede observar en la Plantilla 3 “Plantilla de Requisitos de Evaluación Caribbean Resorts”, se realizará la medición en 6 atributos de calidad para este caso de estudio. Ellos se describen a continuación:

1. Disposición de los Componentes (1.1.4)

En cuanto a la disposición de los componentes en el presente caso de estudio, vemos que inicialmente en la parte central se encuentra el mapa proporcionado por Google Maps, en el cual poco se sobrepone el de Panoramio que una vez que se selecciona con un clic, muestra sobre el mapa la opción de ver fotografías relacionadas al lugar escogido, aunque no es claro si todas las fotografías corresponden al lugar esperado, cada uno de los cuadrados representa una fotografía, que al ser seleccionada agranda la imagen, sin embargo a veces la imagen queda detrás del

borde superior que contiene el menú con los países que se pueden visitar a través de este Mashup.

Otra sección que se puede observar son detalles del lugar al lado izquierdo mediante una sección que puede mostrarse o esconderse según la necesidad, siendo esto muy práctico para el usuario ya que no va a tener siempre la sección de detalles superpuesta en el mapa.

El componente de detalles al momento de recogerse queda muy cómodamente ubicado en la parte inferior, sin presentar ningún problema al usuario.

Así también tenemos un control de búsqueda en la parte inferior que permite ubicar los lugares disponibles, como se comentará luego, es un poco complejo saber a qué se refiere la búsqueda, sin embargo esta no interfiere físicamente con ningún componente.

Cuando ejecutamos el video este se sobrepone a todo el Mashup, pero una vez que termina la reproducción o en cualquier momento se puede esconder sin molestar a los demás componentes.

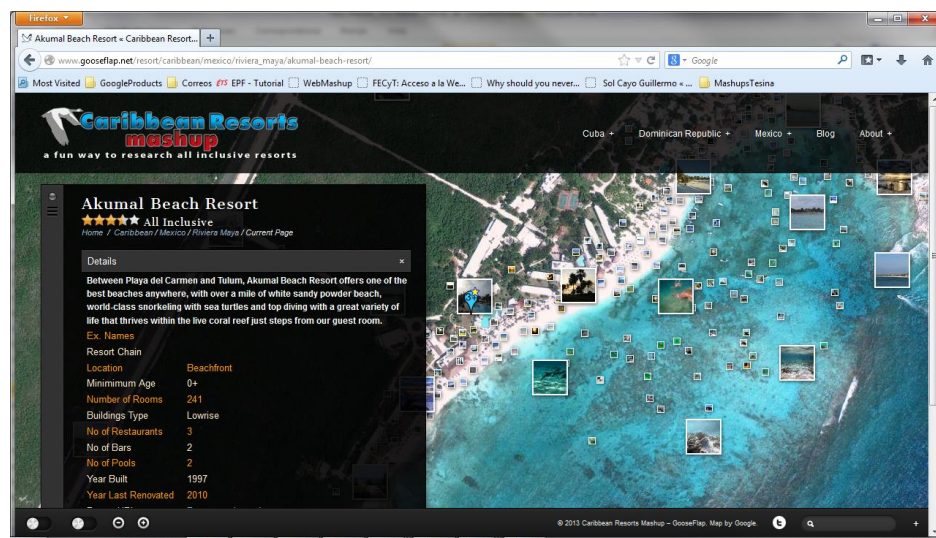


Figura 6-15 Instantánea Mashup Caribbean Resorts.

Como se había explicado en el capítulo del Modelo de calidad, podemos calcular este atributo a través de la métrica:

$$DC = \frac{\text{Número de componentes totalmente visibles}}{\text{Número total de componentes que deberían mostrarse totalmente}}$$

En este caso el número de componentes totalmente visibles son 3 y si bien en cierto momento el mapa es solapado por un cuadro de detalles, este cuadro puede ser minimizado, lo que no hace que represente un problema, por tanto aplicando la métrica tenemos:

$$DC = \frac{4}{5} = 0.8$$

La interpretación de la métrica es $0 \leq valor \leq 1$ en donde el valor más cercano a 1 es el mejor.

De acuerdo al grado de impacto de este atributo en la usabilidad del Mashup se establecen los siguientes umbrales:

Problema mayor de usabilidad:	$0 \leq valor < 0.35$
Problema medio de usabilidad:	$0.35 \leq valor < 0.75$
Problema menor de usabilidad:	$0.75 \leq valor < 1$
Sin ningún problema de usabilidad:	1

Para este caso se evalúa uno de los momentos iniciales del Mashup, habiendo un problema menor de usabilidad en este.

ID	001
Descripción	El componente con las fotografías de Panoramio, queda en ocasiones escondido detrás del menú superior.
Atributo Afectado	1. <i>Facilidad de entendimiento</i> / 1.1. <i>Legibilidad visual</i> / 1.1.4. <i>Disposición de componentes</i> .
Nivel de Criticidad	Menor.
Artefacto(s) evaluado(s)	Todos los componentes del Mashup y el Mashup final visto en totalidad.
Ocurrencias	1
Recomendaciones	Hacer que las fotografías salgan siempre en un solo lugar.
Prioridad	Menor.

Tabla 6-13 Problema de usabilidad 001 detectado

2. Tamaño del Componente (1.1.5)

En el presente caso de estudio se ha tomado de igual manera el estado común e inicial del Mashup, del cual aplicando la fórmula obtenemos:

$$TC = \frac{\text{Número de componentes con tamaño apropiado}}{\text{Número total de componentes}} = \frac{2}{2} = 1$$

En el Mashup de este caso de estudio no se observa ningún problema de usabilidad, ya que el valor obtenido tras la aplicación de la métrica es 1, que según los umbrales no representa problema de usabilidad alguno.

3. Coherencia en la Agrupación de Componentes (1.2.1)

En cuanto a este atributo podemos decir que los componentes se encuentran todos relacionados a la esencia del Mashup, aportando cada uno de ellos al propósito del mismo coherentemente. Al aplicar la fórmula como se puede observar a continuación el valor obtenido es 1, que de acuerdo a los umbrales de referencia establecidos en el Capítulo del Modelo de Usabilidad, no presenta ningún problema.

$$CAC = \frac{\text{Número de componentes coherente}}{\text{Número total de componentes}} = \frac{5}{5} = 1$$

4. Popularidad de los Componentes (1.3.4)

En el caso de la popularidad de los componentes en la cual como se explicó anteriormente, se comprueba si los componentes son conocidos, aceptados y comúnmente usados en la comunidad, para el caso del Mashup Caribbean Resort, tenemos que todos los componentes son conocidos y aceptados:

$$PC = \frac{\text{Número de componentes bien conocidos}}{\text{Número total de componentes}} = \frac{5}{5} = 1$$

Por tanto y según los umbrales presentados en el capítulo 4, no existe ningún problema de usabilidad para este atributo.

5. Actualización de Información entre Componentes (1.4.4)

Como se describió en el modelo de usabilidad este atributo evalúa si los componentes interconectados son actualizados automáticamente, esto es, en el caso de que se refresque por una búsqueda o acción de usuario que requiera dicha actualización en todos los componentes del Mashup, esta actualización se mide de acuerdo a la métrica planteada y en el caso de este ejemplo sería:

$$AIC = \frac{\text{Componentes auto_sincronizados}}{\text{Nro.total componentes que deben autosincronizarse}} = \frac{5}{5} = 1$$

En el caso de este Mashup según el modelo de usabilidad planteado y las métricas recomendadas, no existe problema de usabilidad.

6. Propósito Fácilmente Distinguible de los Componentes (2.1.4)

En este caso de estudio, podemos ver que existen en el Mashup cinco componentes, sin embargo es a primera vista complejo divisarlos, cuando seleccionamos un lugar, aparecen cuadros sobre el mapa, con fotos, es complejo saber si esas fotos corresponden únicamente al lugar seleccionado o están también aquellas de lugares próximos al mismo. De igual manera si queremos ver el video del lugar debemos entrar a los detalles del lugar y en Galería encontraremos las reproducciones, el problema es que no es claro el mensaje que se quiere dar, cuando se hablan de galerías generalmente se vincula a fotos mas no a videos y por último en el componente inferior hay opción de búsqueda, sin embargo no es claro que se puede buscar. Por tanto es difícil encontrar el propósito de los componentes.

$$PFD = \frac{\text{Nro.componentes fácilmente distinguibles}}{\text{Nro.total de componentes}} = \frac{0}{5} = 0$$

Teniendo en cuenta que únicamente el componente del mapa muestra un propósito claro siempre y cuando no se tome en cuenta el control de búsqueda, el componente de los detalles al contener la lupa para búsquedas, indica dificultades porque la búsqueda parece formar parte de los detalles del lugar mas no del lugar como es en realidad, Panoramio no especifica claramente si las fotografías son o no del lugar y el componente de YouTube, está prácticamente escondido, debiendo navegarse mucho para hallarlo entonces vemos que existe un problema mayor de usabilidad en este Mashup, en cuanto a este atributo.

Los umbrales tomados en cuenta para esta valoración fueron:

Problema mayor de usabilidad: $0 \leq \text{valor} < 0.35$

Problema medio de usabilidad: $0.35 \leq \text{valor} < 0.75$

Problema menor de usabilidad: $0.75 \leq \text{valor} < 1$

Sin ningún problema de usabilidad: 1

Mostrándose claramente el problema de usabilidad detectado.

ID	002
Descripción	1. Cuando seleccionamos un lugar, aparecen cuadros sobre el mapa, con fotos, es complejo saber si esas fotos corresponden únicamente al lugar seleccionado o están también aquellas de lugares próximos al mismo. 2. Si queremos ver el video del lugar debemos entrar a los detalles del lugar y en Galería encontraremos las reproducciones, el problema es que no es claro el mensaje que se quiere dar, cuando se hablan de galerías generalmente se vincula a fotos mas no a videos, quedando este componente prácticamente perdido en el Mashup. 3. En el componente inferior hay opción de búsqueda, sin embargo no es claro qué es lo que se puede buscar, este forma parte del mapa pero es ambiguo ya que no se sabe si es del mapa o de los detalles la búsqueda que se puede hacer.
Atributo Afectado	2. <i>Facilidad de aprendizaje</i> / 2.1. <i>Predictibilidad</i> / 2.1.4. <i>Propósito fácilmente distinguible de los componentes</i> .
Nivel de Criticidad	Mayor.
Artefacto(s) evaluado(s)	Todos los componentes del Mashup y el Mashup final.
Ocurrencias	5
Recomendaciones	Reagrupar de una forma más amigable los componentes, presentar más cuadros de texto explicativos y rótulos más claros.
Prioridad	Mayor.

Tabla 6-14

Problema de usabilidad 006 detectado

6.2.3. Evaluación Skypicker

Aquí se evaluará la usabilidad del Mashup Skypicker, teniéndose en cuenta los atributos relacionados directamente con el carácter composicional del Mashup.

A continuación y con el fin de mejorar la explicación de la aplicación se listan los componentes del Mashup en cuestión:

1. Componente para opciones de búsqueda.
2. API Google Maps.
3. Componente de datos de vuelos Low-Cost

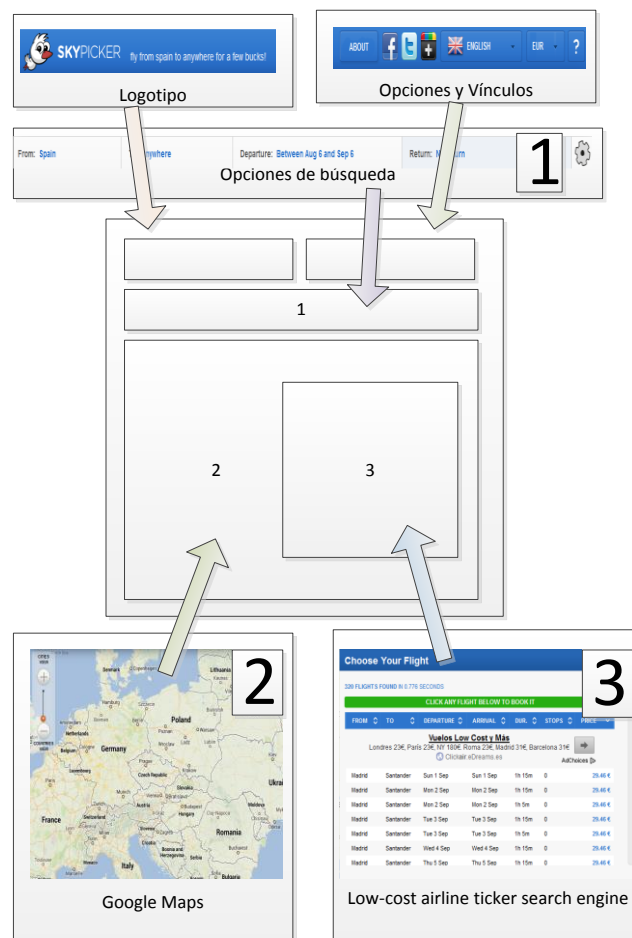


Figura 6-16 Disposición de componentes en Skypicker

Fase1: Especificación de los Requisitos de la Evaluación:

En esta fase se establece el propósito de la evaluación; los perfiles relacionados con el Mashup, el estado del desarrollo en el que se enfocará la evaluación, el contexto de uso y la selección de los atributos de usabilidad. Todos estos aspectos vienen reflejados en el documento requisitos de evaluación.

a) Establecimiento del Propósito de la Evaluación de la Usabilidad de los Mashups

El objetivo es realizar una evaluación de la usabilidad sumativa de la aplicación en cuestión, es decir, se evaluará una aplicación que ya ha sido construida anteriormente, el Mashup está desarrollado con anterioridad y está disponible en la Web como una herramienta terminada.

b) Especificación de los Requisitos de Usabilidad del Mashup

Los requisitos de calidad que se requiere que cumpla la aplicación se ha supuesto para ejemplificar que son los esperados tanto por el desarrollador como por el usuario final, siendo estos dos los principales stakeholders envueltos en el proceso de evaluación descrito aquí.

Adicionalmente se seleccionarán los atributos de usabilidad específicos para Mashups, ya que se quiere remarcar especialmente este tipo de aplicaciones, para ello tendremos en cuenta ciertos atributos de usabilidad comentados y descritos en el Modelo de Usabilidad para Mashups y directamente vinculados con el carácter composicional de estas aplicaciones.

Los atributos a medirse son los siguientes:

Atributos a evaluar	
1	Densidad de la información (1.2.2)
2	Popularidad de los componentes (1.3.4)
3	Presentación de controles en los componentes (1.3.5)
4	Actualización de información entre componentes (1.4.4)

c) Elaboración del Documento de Requisitos de Evaluación

En esta tarea fijaremos un documento que refleje los requisitos de la evaluación, para ello elaboraremos una plantilla, que contendrá la información de las tareas que se realizarán a lo largo del proceso de evaluación.

En base a la Plantilla 1: “Documento de Especificación de Requisitos de Evaluación”, se ha conseguido esta tarea según se muestra en la Plantilla 4: “Plantilla de Requisitos de Evaluación Skypicker”.

Fase 2: Especificación de la Evaluación

En esta fase seleccionaremos aquellos artefactos del Mashup que se desean evaluar y las métricas que serán empleadas. Las métricas seleccionadas van a ser calculadas y se establecerán umbrales para detectar los problemas de usabilidad. Luego definiremos la plantilla del informe para presentar los problemas de usabilidad encontrados. Todos estos aspectos se reflejan en el documento Especificación de la evaluación.

a) Selección de Artefactos a Evaluar

En el caso específico de este caso de estudio, evaluaremos el Mashup final, pero teniendo en cuenta los componentes que se encuentran en el mismo y la forma en que estos han sido unidos para formar la aplicación.

b) Selección de Métricas a Emplear y Definición del Criterio de Decisión para las Métricas de Usabilidad

Atendiendo a la selección de atributos de usabilidad y artefactos a evaluar, se han seleccionado las métricas que serán empleadas en este ejemplo y se presentan en la tabla 6.15.

La prioridad de resolución del problema de usabilidad se evaluará de acuerdo al valor obtenido tras el cálculo de la métrica.

c) Definición del criterio de normalización y agregación

En el caso de este caso de estudio, no definiremos aquí los criterios de normalización y agregación, puesto que estamos evaluando un Mashup en su fase de uso mas no de composición, por tanto no haría falta esta tarea.

d) Definición de la plantilla para el informe de usabilidad

Para la presentación de problemas de usabilidad se usa una plantilla propuesta en el capítulo que especifica el método de evaluación.

Atributo	Métrica	Umbrales
1.2.2. Densidad de la información (DI)	Escala de Likert: La información se muestra en una forma adecuada y sin sobrecargar la página. 1. Totalmente de acuerdo. 2. Ni en acuerdo, ni en desacuerdo. 3. Totalmente en desacuerdo.	En este caso si en la evaluación se muestra la opción 3, se presenta un problema de usabilidad; en el caso de la opción 2, se presenta un problema menor de usabilidad; y por último en primer caso no existe problema ninguno de usabilidad.
1.3.4. Popularidad de los componentes (PC)	$\frac{\text{Nro. de componentes bien conocidos}}{\text{Nro. total de componentes}}$	Problema medio de usabilidad: $0 \leq \text{valor} < 0.5$ Problema menor de usabilidad: $0.5 \leq \text{valor} < 1$ Sin ningún problema de usabilidad: 1
1.3.5. Presentación de controles en componentes (PCC)	$\frac{\text{Controles usuales}}{\text{Nro. componentes que presentan controles}}$	Problema medio de usabilidad: $0 \leq \text{valor} < 0.7$ Problema menor de usabilidad: $0.7 \leq \text{valor} < 1$ Sin ningún problema de usabilidad: 1
1.4.4. Actualización de información entre componentes (AIC)	$\frac{\text{Componentes auto_sincronizados}}{\text{Nro. total comp. que deben autosincronizarse}}$	Problema mayor de usabilidad: $0 \leq \text{valor} < 0.35$ Problema medio de usabilidad: $0.35 \leq \text{valor} < 0.75$ Problema menor de usabilidad: $0.75 \leq \text{valor} < 1$ Sin ningún problema de usabilidad: 1
3.5.3. Facilidad para instalar software adicional (FIS)	$\frac{\text{Nro. Comp. probl. ejecución x soft adic.}}{\text{Nro. comp. que req. software adicional}}$	Problema mayor de usabilidad: $0 \leq \text{valor} < 0.35$ Problema medio de usabilidad: $0.35 \leq \text{valor} < 0.75$ Problema menor de usabilidad: $0.75 \leq \text{valor} < 1$ Sin ningún problema de usabilidad: 1
3.5.4. Compatibilidad con la velocidad de conexión (CVC)	$\frac{\text{Nro. Comp. con ejecución lenta}}{\text{Nro. total de componentes}}$	Problema mayor de usabilidad: $0 \leq \text{valor} < 0.35$ Problema medio de usabilidad: $0.35 \leq \text{valor} < 0.75$ Problema menor de usabilidad: $0.75 \leq \text{valor} < 1$ Sin ningún problema de usabilidad: 1

Tabla 6-15 Definición de Métricas de Calidad

Fase 3: Diseño de la evaluación

En esta fase se indican todas aquellas restricciones de índole técnica que condicionan la evaluación, procediéndose a elaborar el plan de evaluación.

a) Definición de Restricciones

Las principales restricciones con las que se cuenta al momento de realizar la evaluación de este Mashup están relacionadas al hecho de estar evaluando un Mashup que al no ser elaborado directamente por el interesado en realizar la evaluación, no se cuenta con el código ni con la forma de desarrollar directa, para poder luego aplicar las medidas correctivas oportunas, otra restricción también es que se está realizando una evaluación en fase final, no pudiendo observarse el proceso de evaluación en el estado de composición mismo del Mashup, lo que sería de utilidad con el fin ilustrativo de este trabajo, sin embargo se ha tratado de alcanzar un nivel de detalle máximo a fin de cubrir todos los atributos posibles del modelo de usabilidad de tal modo que el lector pueda tener una idea clara de cómo utilizar las métricas y cómo realizar el proceso de evaluación en cualquier estado del Mashup.

b) Elaboración del Plan de Evaluación

El plan de evaluación que se seguirá, será el de aplicar el Modelo de Usabilidad de Mashups, para ir evaluando el caso de estudio propuesto desde una visión global así como también si hace falta, aislando los componentes, es decir “fragmentando” en artefactos de software consistentes en cada componente como una caja negra para evaluar lo que se necesite desde la perspectiva individual de cada componente y cómo este influye en el producto final.

Fase 4: Ejecución de la Evaluación e Informe de Usabilidad

En esta fase se recibe la especificación detallada del plan de evaluación de la calidad así como los requisitos de calidad de la evaluación que se tienen ya disponibles, así también ya se han especificado las métricas para evaluar los artefactos y de pueden obtener los resultados para rellenar los informes de usabilidad correspondientes. Las tareas de esta fase son:

a) Evaluación de los Artefactos, Análisis de Cambios y Elaboración del Informe de Usabilidad

1. Densidad de la información (1.2.2)

Con respecto a la densidad de la información, la misma que tiene que ver con la cantidad de información mostrada al mismo tiempo, según el método de medición referencial, establecimos una escala de Likert en donde realizamos la siguiente pregunta:

La información se muestra en una forma adecuada y sin sobrecargar la página.

1. Totalmente de acuerdo.
2. Ni en acuerdo, ni en desacuerdo.
3. Totalmente en desacuerdo.

En este caso se escogió la opción 1, estableciéndose que no existe un problema de usabilidad.

2. Popularidad de los componentes (1.3.4)

En el caso de la popularidad de los componentes en la cual como se explicó anteriormente, se comprueba si los componentes son conocidos, aceptados y comúnmente usados en la comunidad, para el caso de Skypicker tenemos que dos de los tres componentes son usados comúnmente, el componente de la elección de los lugares y las fechas de ida y regreso y el de Google Maps, por lo tanto medimos de acuerdo a la métrica:

$$PC = \frac{\text{Número de componentes bien conocidos}}{\text{Número total de componentes}} = \frac{2}{3} = 0.67$$

Para este atributo, debemos tener en cuenta que no necesariamente el hecho de que el componente no sea muy conocido pueda afectar a la usabilidad, ya que un componente no tan conocido puede ser muy fácil e intuitivo de usar, sin embargo el usar componentes bien conocidos puede aportar al éxito del Mashup, ya que son componentes aceptados por la comunidad de usuarios que no necesitan un aprendizaje inicial y que ayudan a darle al Mashup una aceptación más rápida. Por esta razón los rangos establecidos son diferentes a los que hemos venido utilizando en atributos anteriores, teniendo:

Problema medio de usabilidad: $0 \leq \text{valor} < 0.5$
 Problema menor de usabilidad: $0.5 \leq \text{valor} < 1$
 Sin ningún problema de usabilidad: 1

Por tanto en Skypicker y de acuerdo a este atributo observamos un problema menor de usabilidad al tener el 0.67 como resultado del cálculo.

ID	001
Descripción	El componente de despliegue de resultados no es popular.
Atributo Afectado	1. Facilidad de entendimiento / 1.3. Familiaridad / 1.3.4. Popularidad de los componentes.
Nivel de Criticidad	Menor.
Artefacto(s) evaluado(s)	Componentes del Mashup Skypicker y el Mashup global.
Ocurrencias	1
Recomendaciones	En este caso se puede recomendar la búsqueda (si existe) de algún componente que despliegue resultados y que sea comúnmente utilizado en este tipo de aplicaciones.
Prioridad	Baja.

Tabla 6-16

Problema de usabilidad 001 detectado

3. Presentación de Controles en Componentes (1.3.5)

Como se había indicado en el Capítulo 4, en este atributo se toma en cuenta si los controles en los componentes son usuales, midiendo este atributo a través de la relación existente entre los componentes con controles usuales y el número total de componentes que presentan controles.

$$PCC = \frac{\text{Controles usuales}}{\text{Nro.Total de Componentes que presentan controles}} = \frac{2}{3} = 0.67$$

En el caso de Skypicker, se presentan en dos de los 3 componentes controles muy usados en el campo informático, como una herramienta de zoom del mapa y así también las flechas hacia arriba y hacia abajo en el componente de despliegue de resultados. El componente superior presenta un problema ya que no es claro que se puede escribir sobre el lugar de la ciudad que se desea buscar y en “Departure” y “Return”, fuera apropiado poner un pequeño icono de calendario a fin de conocer de un solo vistazo que se tratan de fechas. Los umbrales elegidos en este caso son:

Problema medio de usabilidad: $0 \leq \text{valor} < 0.7$
 Problema menor de usabilidad: $0.7 \leq \text{valor} < 1$
 Sin ningún problema de usabilidad: 1

En este caso de estudio se ve un problema medio de usabilidad por tanto, de acuerdo al resultado obtenido y la posterior comprobación de los umbrales.

ID	002
Descripción	El componente superior no presenta controles usuales.
Atributo Afectado	1. <i>Facilidad de entendimiento</i> / 1.3. <i>Familiaridad</i> / 1.3.5. <i>Presentación de controles en componentes</i> .
Nivel de Criticidad	<i>Medio.</i>
Artefacto(s) evaluado(s)	<i>Componentes del Mashup.</i>
Ocurrencias	1
Recomendaciones	Ubicar pequeños íconos al lado de los cuadros de ingreso de texto que hagan comprender de una forma más eficiente de qué se trata esa entrada.
Prioridad	<i>Media.</i>

Tabla 6-17

Problema de usabilidad 002 detectado.

4. Actualización de Información entre Componentes (1.4.4)

Como se describió en el modelo de usabilidad este atributo evalúa si los componentes interconectados son actualizados automáticamente, esto es, en el caso de que se refresque por una búsqueda o acción de usuario que requiera dicha actualización en todos los componentes del Mashup, esta actualización se mide de acuerdo a la métrica planteada y en el caso de este ejemplo sería:

$$AIC = \frac{\text{Componentes auto_sincronizados}}{\text{Nro.total componentes que deben autosincronizarse}} = \frac{2}{3} = 0.67$$

De acuerdo al grado de impacto de este atributo en la usabilidad del Mashup se establecen los siguientes umbrales:

Problema mayor de usabilidad: $0 \leq \text{valor} < 0.35$
 Problema medio de usabilidad: $0.35 \leq \text{valor} < 0.75$
 Problema menor de usabilidad: $0.75 \leq \text{valor} < 1$
 Sin ningún problema de usabilidad: 1

En este caso, el problema de usabilidad es medio, ya que el Mapa no se ve actualizado en la búsqueda, sería interesante remarcar una ruta desde y hasta cuando se hace una búsqueda. Al tratarse de avión, se podrían tomar líneas rectas según los vuelos y escalas para tener una visión rápida del número de escalas y la distancia a recorrer.

ID	003
Descripción	El mapa no se actualiza.
Atributo Afectado	1. <i>Facilidad de entendimiento</i> / 1.4. <i>Reducción de esfuerzo</i> / 1.4.4. <i>Actualización de información entre componentes.</i>
Nivel de Criticidad	<i>Medio.</i>
Artefacto(s) evaluado(s)	<i>Componentes que forman parte del Mashup</i>
Ocurrencias	1
Recomendaciones	Hacer una especie de ruta (aproximada) del vuelo para tener idea de los aeropuertos cuando se dan escalas y de la distancia entre ciudades o países a recorrer.
Prioridad	<i>Media.</i>

Tabla 6-18

Problema de usabilidad 003 detectado

6.3. Conclusiones de la Aplicación del Método de Evaluación

La aplicación del Método de Evaluación propuesto, nos ha permitido obtener los resultados deseados, ya que mediante este Método conseguimos evaluar los artefactos deseados, mostramos los problemas hallados, aplicamos las métricas correspondientes y sugerimos soluciones.

Este capítulo ha mostrado, cómo se aplica el método de evaluación de la usabilidad de Mashups propuesto en este trabajo. El carácter composicional de los Mashups, hace de este tipo de aplicaciones una familia, que si bien forma parte de las aplicaciones Web generales, constituye un tipo muy particular de aplicaciones Web, con características que conllevan la utilización de herramientas más especializadas.

El modelo de usabilidad propuesto, ha servido de base para el método de evaluación propuesto.

Aquí, no solamente hace falta ver la aplicación en conjunto, sino descomponerla y evaluarla por partes, hace también falta ver cómo esas partes se sincronizan y trabajan en conjunto.

Es necesario así mismo, el tener en cuenta que al ser componentes provistos por terceros, estos pueden fallar, ellos pueden estar no disponibles o desaparecer, lo que hace necesario tener en cuenta varios aspectos que ayuden a solventar este tipo de problemas.

Por medio de la aplicación de casos de estudio reales, se ha podido dar una retroalimentación al modelo, mejorarlo y además descubrir aspectos que a simple vista son difíciles de divisar. Además se tomaron la mayoría de atributos y se aplicaron, a fin de conocer cómo podrían ser usadas las métricas, los umbrales y en definitiva todos los pasos que engloba la evaluación.

Cabe señalar que existen atributos que no fueron tomados en cuenta, ya sea porque forman parte de las aplicaciones Web en general y se puede observar su aplicación en el trabajo [19] o porque se deben evaluar en otros estados del ciclo de vida de los Mashups, siendo muy complejo evaluarlos en productos terminados.

Capítulo 7. Conclusiones y Trabajos Futuros

En este capítulo se recogen las conclusiones generales de este trabajo, los trabajos futuros propuestos con miras a la continuación de esta investigación y las publicaciones realizadas como resultados de esta tesina.

7.1. Conclusiones

Los Mashups son aplicaciones Web formadas a partir de componentes y/o recursos provistos por terceros, siendo muy importantes a la hora de desarrollar aplicaciones que solventen un problema puntual y que generalmente tengan un tiempo de vida corto.

La motivación principal de esta tesina de Master, es el contar con un método que permita la evaluación de la usabilidad de este tipo de aplicaciones Web, ya que existen propuestas que se dirigen a las aplicaciones Web en general [19], pero que no abarcan los aspectos claves de composicionalidad de este tipo de aplicaciones Web en particular.

Teniendo esto en consideración, era necesario el contar con un método de Evaluación que permita tener en cuenta cada una de las características propias de los Mashups, permitiendo al usuario y al desarrollador, tener guías para alcanzar un buen grado de usabilidad de sus aplicaciones.

Se ha tenido en cuenta que se pueden evaluar los componentes por separado, ya que una de las fases del ciclo de vida en el desarrollo de los Mashups es la selección de componentes, esta evaluación de componentes debería hacerse al estilo “caja negra”, sin necesidad de detenerse en detalles de implementación de los componentes, sino más bien en el rol que estos juegan en el Mashup final y cómo estos afectan a su desempeño.

Otro aspecto a tener en consideración es la usabilidad del producto como consecuencia del proceso de composición, prestando atención a aspectos claves al momento del desarrollo de los Mashups.

Luego también se ha tenido en cuenta el producto final, al cual incluso se han dirigido los casos de estudio, y en el que se ve globalmente a la aplicación, su funcionalidad y cómo su composicionalidad afecta a su usabilidad.

Según los objetivos planteados en el primer capítulo:

1. Revisar las tecnologías asociadas a la Web 2.0 y que intervengan en una manera directa con los Mashups.
2. Realizar un mapeo sistemático sobre el estado actual de los Mashups, que permita conocer los aspectos de composicionalidad y calidad de este tipo de productos de software.
3. Analizar los modelos de calidad existentes y que puedan servir de base para la elaboración de un modelo específico para la usabilidad de los Mashups.
4. Crear un modelo de usabilidad de Mashups, que permita al evaluador y por qué no al compositor a evaluar y crear respectivamente a fin de conseguir productos altamente usables.

5. Definir un método de evaluación de la usabilidad de los Mashups que utilice el modelo de calidad propuesto.
6. Ejemplificar por medio de casos de estudio la metodología planteada.

Se puede decir que todos y cada uno de ellos han sido abordados y satisfechos:

El objetivo 1 fue conseguido mediante un apartado sobre la base tecnológica con todo lo relacionado a los conceptos, técnicas, estándares y métodos existentes para poder entender los demás capítulos elaborados en esta tesina.

Con respecto al objetivo 2, se realizaron dos mapeos sistemáticos, uno sobre la calidad de los Mashups, desde el punto de vista del producto de software y otra sobre los aspectos de composicionalidad de este tipo de aplicaciones.

Tras estos mapeos se llegó a la conclusión de que muy pocos trabajos abordan la usabilidad de los Mashups y que no existe mucho al respecto de la composicionalidad y cómo esta afecta la calidad del producto en los Mashups.

El objetivo 2, también aportó enormemente al entendimiento del estado del arte en tema de Mashups, los grupos de investigación existentes y hacia donde estos van, cuáles son las brechas en la investigación de estos temas y qué falta por recorrer en los mismos.

El objetivo 3, fue abordado con un estudio de los estándares involucrados y cómo estos pueden aplicarse a un tema determinado dentro del área de la ingeniería del software, en este caso en el ámbito de los Mashups. Se estudió el estándar ISO 25000 SQuaRE y sus predecesores, adicionalmente se describió el estándar 25040 que propone un método de evaluación para productos de software.

Luego se elaboró un Modelo de Usabilidad para Mashups, tomando como punto de partida el modelo de usabilidad para aplicaciones Web descrito en [19]. Este modelo fue extendido y adaptado a los Mashups y a la última norma ISO 25000 SQuaRE. Para ellos se descompuso la usabilidad en sub-características y atributos medibles, teniendo en cuenta los criterios composicionales de los Mashups y dotándolo adicionalmente de métricas genéricas asociadas a dichos atributos, que pueden ser operacionalizadas en base al estado del ciclo de vida del Mashup.

Con respecto al objetivo 5, se definió el proceso de evaluación de usabilidad para Mashups, empleando para ello la notación SPEM, contribuyendo a proporcionar una especificación guiada y detallada del método para los evaluadores. Este método de evaluación utiliza como artefacto principal de entrada el Modelo de Usabilidad de Mashups detallado en el capítulo 4, brindando una serie de recomendaciones al desarrollador para mejorar los aspectos de usabilidad, utilizando siempre las plantillas y siguiendo los pasos definidos en el método de evaluación planteado.

Por último el objetivo 6 se cumplió, aplicando el método de evaluación a tres casos de estudio que muestran los pasos a seguir definidos en el método de evaluación y que permiten evaluar la usabilidad de estas aplicaciones y a su vez brindar las recomendaciones que permiten mejorar las herramientas en cuestión.

Como se puede observar los objetivos de la tesina fueron abarcados en su totalidad y vale la pena destacar que a nivel académico se ha logrado un claro entendimiento del tema, de cómo se puede contribuir a la calidad tomando como base estándares y cómo esta resulta una forma clara de presentar un modelo y un proceso de evaluación de usabilidad que al estar definido con una notación conocida como lo es SPEM puede contribuir claramente y sin ambigüedades a evaluar productos Mashups, pero más allá de ello, se puede extender a aplicaciones de cualquier tipo, ayudando a desarrolladores y personas involucradas en la creación, a obtener productos de calidad.

7.2. Trabajos Futuros

El trabajo presentado, constituye una primera aproximación hacia un método de evaluación de usabilidad Web con carácter composicional, este puede ser aplicado posteriormente a otros tipos de aplicaciones que según los avances tecnológicos están apareciendo, como son aplicaciones en diferentes superficies o dispositivos que tengan características de composicionalidad.

Se plantea también la necesidad de una contribución en la cual se involucre la calidad y la composicionalidad y cómo estas dos características trabajando en conjunto influyen en la usabilidad de los Mashups.

Otro aspecto necesario para realizarlo en un futuro es el refinamiento del modelo de usabilidad mediante la evaluación de un conjunto más amplio de Mashups.

Se realizará en lo posterior una validación empírica del método propuesto a través de familias de experimentos controlados, donde el método sea evaluado objetivamente acorde a su efectividad y eficiencia; y subjetivamente a su facilidad de uso y satisfacción.

Se ve también necesaria para futuros trabajos la creación de una herramienta para dar soporte al método propuesto automatizando la mayor parte posible del mismo.

Otro trabajo futuro sería la propuesta de guías de composición basadas en los resultados obtenidos tras las evaluaciones.

Se espera también realizar una contribución con un acercamiento más dirigido a la industria, que recoja las necesidades puntuales de la misma y que involucre la contribución de la academia a fin de lograr un trabajo conjunto en el cual la investigación sea desde ambos frentes y contribuya de una forma pragmática a la mejora de la usabilidad en los Mashups.

7.3. Publicaciones Relacionadas

Durante el desarrollo de esta tesina de máster, se han realizado dos contribuciones en modo de publicaciones que cubren las aportaciones más relevantes de esta investigación. Estas publicaciones fueron sometidas a un proceso de revisión por pares y consecuentemente aceptadas en conferencias y talleres internacionales. A continuación se presentan estas y se detallan qué parte de la tesina está vinculada a su contenido:

- Insfrán, E.; Cedillo, P.; Fernández, A.; Abrahão, S.; Matera, M.: “Evaluating the Usability of Mashups Applications”, 8th International Conference on the Quality of Information and Communications Technology (**QUATIC 2012**), Lisbon, Portugal, pp: 323-326.

En este artículo se presentó una idea general sobre la evaluación de la usabilidad de los Mashups (Capítulo 5), y una primera versión del Modelo de Usabilidad (Capítulo 4) dirigido a este tipo de aplicaciones.

- Cedillo, P.; Fernández, A.; Insfrán, E.; Abrahão, S.: “Quality of Web Mashups: A Systematic Mapping Study”, 4th International Workshop on Quality in Web Engineering, Aalborg, North Denmark (**QWE 2013**), co-located with the 13th International Conference on Web Engineering (ICWE 2013).

En este artículo se presentó un mapeo sistemático sobre la calidad de los Mashups, (Sección 3.2), con esto se definió el estado actual obre la calidad de los Mashups.

REFERENCIAS

- [1] Aghaee, S.; Pautasso, C.: *"An Evaluation of Mashup Tools Based on Support for Heterogeneous Mashup Components."*; Harth, A. and Koch, N. 11th international conference on Current Trends in Web Engineering. pp. 1–12 , Paphos, Cyprus (2012).
- [2] Aghaee, S.; Pautasso, C.: *"End-User Programming for Web Mashups."*; Harth, A. and Koch, N. Current Trends in Web Engineering. pp. 347–351 Springer Berlin Heidelberg (2012).
- [3] Alkhalifa, E.: *"The Future of Enterprise Mashups"*; Business Insights, E-Strategies for Resource Management Systems, (2009).
- [4] Ardito, C.; Costabile, M.F.; Desolda, G.; Matera, M.; Piccinno, A.; Picozzi, M.: *"Composition of situational interactive spaces by end users: a case for cultural heritage."*; Proceedings of the 7th Nordic Conference on Human-Computer Interaction: Making Sense Through Design. pp. 79–88 ACM, New York, NY, USA (2012).
- [5] Bastien, J.; Scapin, D.: *"Ergonomic Criteria for the Evaluation of Human-Computer Interfaces"*; TR.156. INRIA, (1993).
- [6] Bean, J.: *"SOA and Web Services Interface Design: Principles, Techniques, and Standards."* Elsevier, Burlington (2009).
- [7] Biörnstad, B.; Pautasso, C.: *"Let It Flow: Building Mashups with Data Processing Pipelines."*; Nitto, E. and Ripeanu, M. Service-Oriented Computing - ICSOC 2007 Workshops. pp. 15–28 Springer Berlin Heidelberg (2009).
- [8] Bozzon, A.; Brambilla, M.; Facca, F.M.; Carughu, G.T.: *"A Conceptual Modeling Approach to Business Service Mashup Development."*; Web Services, 2009. ICWS 2009. IEEE International Conference on. pp. 751–758 (2009).
- [9] Brandon, B.: *"Mashups: A Literature Review and Classification Framework"*; Future Internet, 1, 59–87 (2009).
- [10] Calero, C.; Ruiz, J.; Piattini, M.: *"Classifying Web Metrics Using the Web Quality Model"*; Emerald Group Publishing Limited, 29, 3, 227–248 (2005).
- [11] Cappiello, C.; Daniel, F.; Koschmider, A.; Matera, M.; Picozzi, M.: *"A Quality Model for Mashups."*; Auer, S., Díaz, O., and Papadopoulos, G. Web Engineering. pp. 137–151 Springer Berlin Heidelberg (2011).
- [12] Cappiello, C.; Daniel, F.; Matera, M.: *"A Quality Model for Mashup Components."*; Gaedke, M., Grossniklaus, M., and Díaz, O. 9th International Conference on Web Engineering (ICWE). pp. 236–250 , San Sebastian, Spain (2009).
- [13] Cappiello, C.; Daniel, F.; Matera, M.; Pautasso, C.: *"Information Quality in Mashups"*; IEEE Computer Society, 30–38 (2010).
- [14] Cappiello, C.; Matera, M.; Picozzi, M.; Daniel, F.; Fernandez, A.: *"Quality-Aware Mashup Composition: Issues, Techniques and Tools."*; Eighth International Conference on the Quality of Information and Communications Technology (QUATIC). pp. 10–19 , Lisbon, Portugal (2012).
- [15] Cardona, G.P.; Garijo, M.: *"A Conceptual Architecture for Semantic Mash Up Recommender Framework."*; Signal Image Technology and Internet Based Systems (SITIS), 2012 Eighth International Conference on. pp. 1002–1009 (2012).
- [16] Chatti, M.; Jarke, M.; Specht, M.; Schroeder, U.; Dahl, D.: *"Model-Driven Mashup Personal Learning Environments"*; Int. J. Technology Enhanced Learning, 3, 1, 21–39 (2011).
- [17] Costabile, M.F., Fogli, D., Mussio, P., Piccinno, A.: *"End-User Development."* , Dordrecht (2006).

- [18] Fenton, N.E.; Pfleeger, S.L.: *"Software Metrics: A Rigorous and Practical Approach."* PWS Publishing Co., Boston, MA, USA (1998).
- [19] Fernández, A.: *"A Usability Inspection Method for Model-driven Web Development Processes."*; Universitat Politècnica de València (2012).
- [20] Fernandez, A.; Insfrán, E.; Abrahao, S.: *"Integrating a Usability Model into Model-Driven Web Development Processes."*; Web Information Systems Engineering (WISE). pp. 497–510 Springer Berlin Heidelberg, Poznan, Poland (2009).
- [21] Garousi, V.; Mesbah, A.; Betin-Can, A.; Mirshokraie, S.: *"A systematic mapping study of Web application testing"*; Information and Software Technology, 0, - (2013).
- [22] Grammel, L.; Storey, M.-A.: *"The smart internet."*; Presented at the (2010).
- [23] Griffin, E.: *"Foundations of Popfly. Rapid Mashup Development."*, New York (2008).
- [24] Gurram, R.; Mo, B.; Gueldemeister, R.: *"A Web Based Mashup Platform for Enterprise 2.0."*; Hartmann, S., Zhou, X., and Kirchberg, M. Web Information Systems Engineering (WISE). pp. 144–151 Springer Berlin Heidelberg (2008).
- [25] Herbert, M.; Thieme, T.; Zibuschka, J.; Roßnagel, H.: *"Secure Mashup-Providing Platforms - Implementing Encrypted Wiring."*; Harth, A. and Koch, N. 11th international conference on Current Trends in Web Engineering. pp. 99–108 Springer Berlin Heidelberg, Paphos, Cyprus (2011).
- [26] Hoyer, V.; Fischer, M.: *"Market Overview of Enterprise Mashup Tools."*; Bouguettaya, A., Krueger, I., and Margaria, T. Service-Oriented Computing (ICSOC). pp. 708–721 Springer Berlin Heidelberg, Sydney, Australia (2008).
- [27] Huang, J.; Liu, X.; Zhao, Q.; Ma, J.; Huang, G.: *"A browser-based framework for data cache in Web-delivered service composition."*; Service-Oriented Computing and Applications (SOCA), 2010 IEEE International Conference on. pp. 1–8 (2010).
- [28] Ikeda, S.; Nagamine, T.; Kamada, T.: *"Application framework with demand-driven mashup for selective browsing."*; Proceedings of the 10th International Conference on Information Integration and Web-based Applications & Services. pp. 33–40 ACM, New York, NY, USA (2008).
- [29] International Standards Organization: *"ISO 13407. Human Centred Design Process for Interactive Systems,"*; (1999).
- [30] ISO/IEC: *"ISO 9241-210:2010 Ergonomics of human-system interaction,"*; http://www.iso.org/iso/iso_catalogue/catalogue_ics/catalogue_detail_ics.htm?csnumber=52075, (2010).
- [31] ISO/IEC: *"ISO/IEC 25010 Systems and software engineering - Systems and Software Quality Requirements and Evaluation (SQuaRE) -- System and software quality models,"*; (2011).
- [32] ISO/IEC: *"ISO/IEC 9126. Software engineering -- Product quality."* ISO/IEC (2001).
- [33] Jackson, C.; Wang, H.J.: *"Subspace: secure cross-domain communication for Web mashups."*; Proceedings of the 16th international conference on World Wide Web. pp. 611–620 ACM, New York, NY, USA (2007).
- [34] Janiesch, C.; Fleischmann, K.; Dreiling, A.: *"Extending Services Delivery with Lightweight Composition."*; Hartmann, S., Zhou, X., and Kirchberg, M. Web Information Systems Engineering – WISE 2008 Workshops. pp. 162–171 Springer Berlin Heidelberg (2008).
- [35] Janner, T.; Siebeck, R.; Schroth, C.; Hoyer, V.: *"Patterns for Enterprise Mashups in B2B Collaborations to Foster Lightweight Composition and End User Development."*; Web Services, 2009. ICWS 2009. IEEE International Conference on. pp. 976–983 (2009).
- [36] Kitchenham, B.; Charters, S.: *"Guidelines for performing Systematic Literature Reviews in Software Engineering."* (2007).
- [37] Koschmider, A.; Hoyer, V.; Giessmann, A.: *"Quality metrics for mashups."*; Proceedings of the 2010 Annual Research Conference of the South African Institute of Computer

- Scientists and Information Technologists. pp. 376–380 ACM, New York, NY, USA (2010).
- [38] Koschmider, A.; Torres, V.; Pelechano, V.: “*Elucidating the Mashup Hype : Definition, Challenges, Methodical Guide and Tools for Mashups.*”;2nd Workshop on Mashups, Enterprise Mashups and Lightweight Composition on the Web. , Madrid, Spain (2009).
 - [39] Leavin, M.; Shneiderman, B.: “*Research-Based Web Design & Usability Guidelines*”; U.S. Government Printing Office, (2006).
 - [40] Lee, Y.-J.; Kim, J.-S.: “*Automatic Web API Composition for Semantic Data Mashups.*”;Computational Intelligence and Communication Networks (CICN), 2012 Fourth International Conference on. pp. 953–957 (2012).
 - [41] Liu, X.; Zhao, Q.; Huang, G.; Mei, H.; Teng, T.: “*Composing Data-Driven Service Mashups with Tag-Based Semantic Annotations.*”;Web Services (ICWS), 2011 IEEE International Conference on. pp. 243–250 (2011).
 - [42] López, J.; Bellas, F.; Pan, A.; Montoto, P.: “*A Component-Based Approach for Engineering Enterprise Mashups.*”;Gaedke, M.,Grossniklaus, M.,and Díaz, O. Web Engineering. pp. 30–44 Springer Berlin Heidelberg (2009).
 - [43] Di Lorenzo, G.; Hacid, H.; Paik, H.; Benatallah, B.: “*Data integration in mashups*”; SIGMOD Rec., 38, 1, 59–66 (2009).
 - [44] Magazinius, J.; Askarov, A.; Sabelfeld, A.: “*A lattice-based approach to mashup security.*”;Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security. pp. 15–23 ACM, New York, NY, USA (2010).
 - [45] Makki, S.K.; Sangtani, J.: “*Data Mashups & Their Applications in Enterprises.*”;Internet and Web Applications and Services, 2008. ICIW '08. Third International Conference on. pp. 445–450 (2008).
 - [46] Maximilien, E.M.; Wilkinson, H.; Desai, N.; Tai, S.: “*A Domain-Specific Language for Web APIs and Services Mashups.*”;Krämer, B.,Lin, K.-J.,and Narasimhan, P. Service-Oriented Computing – ICSOC 2007. pp. 13–26 Springer Berlin Heidelberg (2007).
 - [47] Mehandjiev, N.; Lecue, F.; Wajid, U.; Namoun, A.: “*Assisted Service Composition for End Users.*”;Web Services (ECOWS), 2010 IEEE 8th European Conference on. pp. 131–138 (2010).
 - [48] Moraga, M.; Calero, C.; Piattini, M.; Diaz, O.: “*Improving a Portlet Usability Model*”; Improving a Portlet Usability Model, 15, 2, 155–177 (2007).
 - [49] Olsina, L.; Lew, P.; Dieser, A.; Rivera, B.: “*Updating quality models for evaluating new generation Web applications*”; J. Web Eng., 11, 3, 209–246 (2012).
 - [50] Olsina, L.; Rossi, G.: “*Measuring Web Application Quality with Web QEM*”; IEEE Multimedia, 9, 4, 20–29 (2002).
 - [51] OMG: “*Software Process Engineering Meta-model v2.0.*”; <http://www.omg.org/technology/documents/formal/spem.htm>.
 - [52] OrangeLabs: “*State of the Art in Mashup tools.*”; (2011).
 - [53] Petersen, K.; Feldt, R.; Mujtaba, S.; Mattsson, M.: “*Systematic mapping studies in software engineering.*”;Proceedings of the 12th international conference on Evaluation and Assessment in Software Engineering. pp. 68–77 British Computer Society, Swinton, UK, UK (2008).
 - [54] Raza, M.; Hussain, F. khadeer; Chang, E.: “*A methodology for quality-based mashup of data sources.*”;Proceedings of the 10th International Conference on Information Integration and Web-based Applications & Services. pp. 528–533 ACM, New York, NY, USA (2008).
 - [55] Riabov, A. V; Boillet, E.; Feblowitz, M.D.; Liu, Z.; Ranganathan, A.: “*Wishful search: interactive composition of data mashups.*”;Proceedings of the 17th international conference on World Wide Web. pp. 775–784 ACM, New York, NY, USA (2008).

- [56] Rosenberg, F.; Khalaf, R.; Duftler, M.; Curbera, F.; Austel, P.: *"End-to-End Security for Enterprise Mashups."*;Baresi, L.,Chi, C.-H.,and Suzuki, J. 7th International Joint Conference, ICSOC-ServiceWave. pp. 389–403 Springer Berlin Heidelberg, Stockholm, Sweden (2009).
- [57] Roy Chowdhury, S.; Daniel, F.; Casati, F.: *"Efficient, Interactive Recommendation of Mashup Composition Knowledge."*;Kappel, G.,Maamar, Z.,and Motahari-Nezhad, H. Service-Oriented Computing. pp. 374–388 Springer Berlin Heidelberg (2011).
- [58] Saeed, A.: *"A Quality-based Framework for Leveraging the Process of Mashup Component Selection."*;University of Gothenburg (2010).
- [59] Dos Santos, C.R.P.; Bezerra, R.S.; Granville, L.Z.; Bertholdo, L.M.; Cheng, W.; Anerousis, N.: *"A data confidentiality architecture for developing management mashups."*;International Symposium on Integrated Network Management (IM). pp. 49–56 , Dublin, Ireland (2011).
- [60] Seffah, A.; Donyaee, M.; Kline, R.; Padda, K.: *"Usability Measurement and Metrics"*; A Consolidated Model. Software Quality Journal, 14, 2, 159–178 (2006).
- [61] Soyulu, A.; Wild, F.; Mödritscher, F.; Desmet, P.; Verlinde, S.; De Causmaecker, P.: *"Mashups and widget orchestration."*;Proceedings of the International Conference on Management of Emergent Digital EcoSystems. pp. 226–234 ACM, New York, NY, USA (2011).
- [62] Stecca, M.; Maresca, M.: *"Mashup Patterns from Service Component Taxonomy."*;24th International Conference on Advanced Information Networking and Applications Workshops (WAINA). pp. 12–17 , Bradford, United Kingdom (2010).
- [63] Tritschler, M.; Kleinfeld, R.; Steglich, S.: *"Enhancing Interoperability in Cross-Platform Enterprise Mashups through Data Aggregation and Extraction."*;13th International Conference on High Performance Computing and Communications (HPCC),. pp. 695–700 , Banff, AB, Canadá (2011).
- [64] Vancea, A.; Grossniklaus, M.; Norrie, M.C.: *"Database-Driven Web Mashups."*;Web Engineering, 2008. ICWE '08. Eighth International Conference on. pp. 162–174 (2008).
- [65] Wang, G.; Liu, J.; Cao, B.; Tang, M.: *"Mashup Service Classification and Recommendation Based on Similarity Computing."*;Cloud and Green Computing (CGC), 2012 Second International Conference on. pp. 621–628 (2012).
- [66] Wohlstadter, E.; Li, P.; Cannon, B.: *"Web Service Mashup Middleware with Partitioning of XML Pipelines."*;Web Services, 2009. ICWS 2009. IEEE International Conference on. pp. 91–98 (2009).
- [67] Xie, L.; Xu, L.; de Vrieze, P.: *"Process modelling in process-oriented enterprise mashups."*;2nd IEEE International Conference on Information Management and Engineering (ICIME). pp. 650–654 , Chengdu, Sichuan, China (2010).
- [68] Yu, J.; Benatallah, B.; Casati, F.; Daniel, F.: *"Understanding Mashup Development."*;IEEE Internet Computing. pp. 44–52 IEEE Educational Activities Department, Piscataway, NJ, USA (2008).
- [69] Zajicek, M.: *"Web 2.0: hype or happiness?"*;Proceedings of the 2007 international cross-disciplinary conference on Web accessibility (W4A). pp. 35–39 ACM, New York, NY, USA (2007).
- [70] Zhao, Q.; Huang, G.; Huang, J.; Liu, X.; Mei, H.: *"A Web-Based Mashup Environment for On-the-Fly Service Composition."*;Service-Oriented System Engineering, 2008. SOSE '08. IEEE International Symposium on. pp. 32–37 (2008).

APENDICE A – BIBLIOGRAFIA MAPEO SISTEMATICO CALIDAD

- [Q1] Barbagallo, D.; Cappiello, C.; Francalanci, C.; Matera, M.; Picozzi, M.: “Informing observers: quality-driven filtering and composition of Web 2.0 sources.” Joint EDBT/ICDT Workshops. pp. 1–8 ACM, New York, NY, USA (2012).
- [Q2] Björnstad, B.; Pautasso, C.: “Let It Flow: Building Mashups with Data Processing Pipelines.” Nitto, E. and Ripeanu, M. 5th International Conference on Service-Oriented Computing - (ICSOC). pp. 15–28 Springer Berlin Heidelberg, Vienna, Austria (2007).
- [Q3] Bozzon, A.; Brambilla, M.; Facca, F.M.; Carughu, G.T.: “A Conceptual Modeling Approach to Business Service Mashup Development.” IEEE International Conference on Web Services (ICWS). pp. 751–758 , Los Angeles, CA, USA (2009).
- [Q4] Cappiello, C.; Daniel, F.; Koschmider, A.; Matera, M.; Picozzi, M.: “A Quality Model for Mashups.” Auer, S.,Díaz, O.,and Papadopoulos, G. 9th International Conference, (ICWE). pp. 137–151 Springer Berlin Heidelberg, San Sebastián, Spain (2011).
- [Q5] Cappiello, C.; Daniel, F.; Matera, M.: “A Quality Model for Mashup Components.” Gaedke, M.,Grossniklaus, M.,and Díaz, O. 9th International Conference on Web Engineering (ICWE). pp. 236–250 , San Sebastian, Spain (2009).
- [Q6] Cappiello, C.; Daniel, F.; Matera, M.; Pautasso, C.: “Information Quality in Mashups” ; IEEE Computer Society, 30–38 (2010).
- [Q7] Cappiello, C.; Matera, M.; Picozzi, M.; Daniel, F.; Fernandez, A.: “Quality-Aware Mashup Composition: Issues, Techniques and Tools.” 8th International Conference on the Quality of Information and Communications Technology (QUATIC). pp. 10–19 , Lisbon, Portugal (2012).
- [Q8] Chu, D.; Liao, Q.; Zhao, J.: “Open identity management framework for mashup.” 2nd Symposium on Web Society (SWS). pp. 378–382 , Beijing, China (2010).
- [Q9] Durdik, Z.; Drawehn, J.; Herbert, M.: “Towards automated service quality prediction for development of enterprise mashups.” 5th International Workshop on Web APIs and Service Mashups. pp. 6:1–6:8 ACM, New York, NY, USA (2011).
- [Q10] Elmeleegy, H.; Ivan, A.; Akkiraju, R.; Goodwin, R.: “Mashup Advisor: A Recommendation Tool for Mashup Development.” IEEE International Conference on Web Services (ICWS). pp. 337–344 , Beijing, China (2008).
- [Q11] Gao, P.; Han, H.: “From toys to products: a step towards supporting the robust reuse and integration on the Web.” 6th International Conference on Ubiquitous Information Management and Communication. pp. 78:1–78:9 ACM, New York, NY, USA (2012).
- [Q12] Govardhan, S.; Feuerlicht, G.: “Itinerary Planner: A Mashup Case Study.” Nitto, E. and Ripeanu, M. 5th International Conference on Service-Oriented Computing - (ICSOC). pp. 3–14 Springer Berlin Heidelberg, Vienna, Austria (2007).

- [Q13] Gurram, R.; Mo, B.; Gueldemeister, R.: "A Web Based Mashup Platform for Enterprise 2.0." Hartmann, S., Zhou, X., and Kirchberg, M. Web Information Systems Engineering (WISE). pp. 144–151 Springer Berlin Heidelberg, Auckland, New Zealand (2008).
- [Q14] Haque, M.M.; Ahsan, A.; Yu, W.; Andersen, A.; Karlsen, R.: "Peer-to-peer orchestration of Web mashups." 14th International Conference on Information Integration and Web-based Applications & Services. pp. 294–298 ACM, New York, NY, USA (2012).
- [Q15] Hoyer, V.; Fischer, M.: "Market Overview of Enterprise Mashup Tools." Bouguettaya, A., Krueger, I., and Margaria, T. Service-Oriented Computing (ICSOC). pp. 708–721 Springer Berlin Heidelberg, Sydney, Australia (2008).
- [Q16] Jackson, C.; Wang, H.J.: "Subspace: secure cross-domain communication for Web mashups." 16th international conference on World Wide Web. pp. 611–620 ACM, New York, NY, USA (2007).
- [Q17] De Keukelaere, F.; Bhola, S.; Steiner, M.; Chari, S.; Yoshihama, S.: "SMash: secure component model for cross-domain mashups on unmodified browsers." 17th international conference on World Wide Web. pp. 535–544 ACM, New York, NY, USA (2008).
- [Q18] Koschmider, A.; Hoyer, V.; Giessmann, A.: "Quality metrics for mashups." Proceedings of the 2010 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists. pp. 376–380 ACM, New York, NY, USA (2010).
- [Q19] Liu, D.; Li, N.; Pedrinaci, C.; Kopecký, J.; Maleshkova, M.; Domingue, J.: "An Approach to Construct Dynamic Service Mashups Using Lightweight Semantics." Harth, A. and Koch, N. Doctoral Symposium, and Tutorials, Held (ICWE). pp. 13–24 Springer Berlin Heidelberg, Paphos, Cyprus (2012).
- [Q20] Liu, X.; Jiang, N.; Zhao, Q.; Huang, G.: "iSocialMash: Convergence of social networks and services composition on a mashup framework." IEEE International Conference on Service-Oriented Computing and Applications (SOCA). pp. 1–6 , California, Irvine, USA (2011).
- [Q21] Liu, X.; Zhao, Q.; Huang, G.; Mei, H.; Teng, T.: "Composing Data-Driven Service Mashups with Tag-Based Semantic Annotations." IEEE International Conference on Web Services (ICWS). pp. 243–250 , Washington DC, USA (2011).
- [Q22] López, J.; Bellas, F.; Pan, A.; Montoto, P.: "A Component-Based Approach for Engineering Enterprise Mashups." Gaedke, M., Grossniklaus, M., and Díaz, O. 9th International Conference, (ICWE). pp. 30–44 Springer Berlin Heidelberg, San Sebastián, Spain (2009).
- [Q23] Luo, X.; Song, M.; Liu, L.: "A solution for Mashup Platform based on SOA." Joint Conferences on Pervasive Computing (JCPC). pp. 483–488 , Tamsui, Taipei, Taiwan (2009).

- [Q24] Magazinius, J.; Askarov, A.; Sabelfeld, A.: "A lattice-based approach to mashup security." 5th ACM Symposium on Information, Computer and Communications Security. pp. 15–23 ACM, New York, NY, USA (2010).
- [Q25] Olsina, L.; Lew, P.; Dieser, A.; Rivera, B.: "Updating quality models for evaluating new generation Web applications" ; J. Web Eng., 11, 3, 209–246 (2012).
- [Q26] Picozzi, M.; Rodolfi, M.; Cappiello, C.; Matera, M.: "Quality-Based Recommendations for Mashup Composition." Daniel, F. and Facca, F. 10th International Conference on Web Engineering (ICWE). pp. 360–371 Springer Berlin Heidelberg, Vienna, Austria (2010).
- [Q27] Raza, M.; Hussain, F. khadeer; Chang, E.: "A methodology for quality-based mashup of data sources." 10th International Conference on Information Integration and Web-based Applications & Services. pp. 528–533 ACM, New York, NY, USA (2008).
- [Q28] Riabov, A. V; Boillet, E.; Feblowitz, M.D.; Liu, Z.; Ranganathan, A.: "Wishful search: interactive composition of data mashups." Proceedings of the 17th international conference on World Wide Web. pp. 775–784 ACM, New York, NY, USA (2008).
- [Q29] Rosenberg, F.; Khalaf, R.; Duftler, M.; Curbera, F.; Austel, P.: "End-to-End Security for Enterprise Mashups." Baresi, L., Chi, C.-H., and Suzuki, J. 7th International Joint Conference, ICSOC-ServiceWave. pp. 389–403 Springer Berlin Heidelberg, Stockholm, Sweden (2009).
- [Q30] Roy Chowdhury, S.; Daniel, F.; Casati, F.: "Efficient, Interactive Recommendation of Mashup Composition Knowledge." Kappel, G., Maamar, Z., and Motahari-Nezhad, H. 9th international conference on Service-Oriented Computing (ICSOC). pp. 374–388 Springer Berlin Heidelberg, Berlin, Germany (2011).
- [Q31] Dos Santos, C.R.P.; Bezerra, R.S.; Granville, L.Z.; Bertholdo, L.M.; Cheng, W.; Anerousis, N.: "A data confidentiality architecture for developing management mashups." International Symposium on Integrated Network Management (IM). pp. 49–56 , Dublin, Ireland (2011).
- [Q32] Smith, R.; Xu, J.: "A survey of personal privacy protection in public service mashups." 6th International Symposium on Service Oriented System Engineering (SOSE). pp. 214–224 , Irving, CA, USA (2011).
- [Q33] Stecca, M.; Fornasa, M.; Dall'Armellina, N.; Maresca, M.: "Event-Driven Mashup Orchestration with Scala." 9th International Conference on Services Computing (SCC),. pp. 531–538 , Honolulu, Hawaii, USA (2012).
- [Q34] Stecca, M.; Maresca, M.: "An Architecture for a Mashup Container in Virtualized Environments." 3rd International Conference on Cloud Computing (CLOUD). pp. 386–393 , Miami, FL, USA (2010).
- [Q35] Stecca, M.; Maresca, M.: "An execution platform for event driven mashups." 11th International Conference on Information Integration and Web-based Applications & Services. pp. 33–40 ACM, Kuala Lumpur, Malaysia (2009).

- [Q36] Treiber, M.; Kritikos, K.; Schall, D.; Dustdar, S.; Plexousakis, D.: "Modeling context-aware and socially-enriched mashups." 3rd and 4th International Workshop on Web APIs and Services Mashups. pp. 2:1–2:8 ACM, New York, NY, USA (2010).
- [Q37] Wohlstadter, E.; Li, P.; Cannon, B.: "Web Service Mashup Middleware with Partitioning of XML Pipelines." IEEE International Conference on Web Services (ICWS). pp. 91–98 , Los Angeles, CA, USA (2009).
- [Q38] Zhao, Q.; Liu, X.; Huang, G.; Huang, J.; Mei, H.: "A Browser-Based Middleware for Service-Oriented Rich Client." International Conference on Service Sciences (ICSS). pp. 22–27 , Rome, Italy (2010).

APENDICE B – BIBLIOGRAFIA MAPEO SISTEMATICO COMPOSICIONALIDAD

- [CO1] Abiteboul, S.; Greenshpan, O.; Milo, T.: "Modeling the mashup space." 10th ACM workshop on Web information and data management. pp. 87–94 ACM, New York, NY, USA (2008).
- [CO2] Aghaee, S.; Pautasso, C.: "End-User Programming for Web Mashups." *Current Trends in Web Engineering*. pp. 347–351 Springer Berlin Heidelberg (2012).
- [CO3] Aghaee, S.; Pautasso, C.: "The mashup component description language." 13th International Conference on Information Integration and Web-based Applications and Services. pp. 311–316 ACM, New York, NY, USA (2011).
- [CO4] De Alwis, B.; Malinga, S.; Pradeeban, K.; Weerasiri, D.; Perera, S.; Nanayakkara, V.: "Mooshabaya: mashup generator for XBaya." 8th International Workshop on Middleware for Grids, Clouds and e-Science. pp. 8:1–8:6 ACM, New York, NY, USA (2010).
- [CO5] Ardito, C.; Costabile, M.F.; Desolda, G.; Matera, M.; Piccinno, A.; Picozzi, M.: "Composition of situational interactive spaces by end users: a case for cultural heritage." 7th Nordic Conference on Human-Computer Interaction: Making Sense Through Design. pp. 79–88 ACM, New York, NY, USA (2012).
- [CO6] Barbagallo, D.; Cappiello, C.; Francalanci, C.; Matera, M.; Picozzi, M.: "Informing observers: quality-driven filtering and composition of Web 2.0 sources." Joint EDBT/ICDT Workshops. pp. 1–8 ACM, New York, NY, USA (2012).
- [CO7] Barhamgi, M.; Ghedira, C.; Benslimane, D.; Tbahrity, S.-E.; Mrissa, M.: "Optimizing DaaS Web Service Based Data Mashups." IEEE International Conference on Services Computing (SCC). pp. 464–471 , Washington DC, USA (2011).
- [CO8] Belaunde, M.; Hassen, S.B.: "Service Mashups Using Natural Language and Context Awareness: A Pragmatic Architectural Design." 15th IEEE International Enterprise Distributed Object Computing Conference Workshops (EDOCW). pp. 404–411 , Helsinki, Finland (2011).
- [CO9] Belaunde, M.; Pinson, F.; Pellen, N.: "Biologeeek, an intelligent system for service mashups tuned for recipe processing and rendering." Workshop on Multimedia for cooking and eating activities. pp. 49–54 ACM, New York, NY, USA (2012).
- [CO10] Bianchini, D.; De Antonellis, V.; Melchiori, M.: "Semantic-driven mashup design." 12th International Conference on Information Integration and Web-based Applications and Services Services. pp. 247–254 ACM, New York, NY, USA (2010).
- [CO11] Björnstad, B.; Pautasso, C.: "Let It Flow: Building Mashups with Data Processing Pipelines." Nitto, E. and Ripeanu, M. 5th International Conference on Service-Oriented Computing - (ICSOC). pp. 15–28 Springer Berlin Heidelberg, Vienna, Austria (2007).

- [CO12] Blau, B.; Neumann, D.; Weinhardt, C.; Lamparter, S.: "Planning and Pricing of Service Mashups." 10th E-Commerce Technology and the Fifth IEEE Conference on Enterprise Computing, E-Commerce and E-Services. pp. 19–26 , Washington, DC, USA (2008).
- [CO13] Cappiello, C.; Matera, M.; Picozzi, M.; Daniel, F.; Fernandez, A.: "Quality-Aware Mashup Composition: Issues, Techniques and Tools." 8th International Conference on the Quality of Information and Communications Technology (QUATIC). pp. 10–19 , Lisbon, Portugal (2012).
- [CO14] Cappiello, C.; Matera, M.; Picozzi, M.; Sprega, G.; Barbagallo, D.; Francalanci, C.: "DashMash: A Mashup Environment for End User Development." Auer, S.,Díaz, O.,and Papadopoulos, G. 11th International Conference on Web Engineering (ICWE). pp. 152–166 Springer Berlin Heidelberg, Paphos, Cyprus (2011).
- [CO15] Cardona, G.P.; Garijo, M.: "A Conceptual Architecture for Semantic Mash Up Recommender Framework." 8th International Conference on Signal Image Technology and Internet Based Systems (SITIS). pp. 1002–1009 , Naples, Italy (2012).
- [CO16] Chu, D.; Liao, Q.; Zhao, J.: "Open identity management framework for mashup." 2nd Symposium on Web Society (SWS). pp. 378–382 , Beijing, China (2010).
- [CO17] Chung, S.; Shin, Y.-M.: "SCE Library Implementation for Creating Networks Converged Mashup Service." 10th International Conference on Advanced Communication Technology (CACT). pp. 1667–1671 , Phoenix Park, Gangwon-Do, Korea (2008).
- [CO18] Cremonesi, P.; Picozzi, M.; Matera, M.: "A comparison of recommender systems for mashup composition." 3th International Workshop on Recommendation Systems for Software Engineering (RSSE). pp. 54–58 , Zurich, Switzerland (2012).
- [CO19] Daniel, F.; Matera, M.: "Mashing Up Context-Aware Web Applications: A Component-Based Development Approach." Bailey, J.,Maier, D.,Schewe, K.-D.,Thalheim, B.,and Wang, X. Web Information Systems Engineering (WISE). pp. 250–263 Springer Berlin Heidelberg, Auckland, New Zealand (2008).
- [CO20] Dorn, C.; Schall, D.; Dustdar, S.: "Context-aware adaptive service mashups." IEEE Asia-Pacific Services Computing Conference (APSCC). pp. 301–306 , Biopolis, Singapore (2009).
- [CO21] Fischer, T.; Bakalov, F.; König-Ries, B.; Nauertz, A.; Welsch, M.: "An Evolutionary Algorithm for Automatic Composition of Information-gathering Web Services in Mashups." 7th IEEE European Conference on Web Services (ECOWS). pp. 39–48 , Los Angeles, CA, USA (2009).
- [CO22] Fox, R.; Cooley, J.; Hauswirth, M.: "Collaborative development of trusted mashups." 12th International Conference on Information Integration and Web-based Applications and Services. pp. 255–262 ACM, New York, NY, USA (2010).

- [CO23] Gao, P.; Han, H.: "From toys to products: a step towards supporting the robust reuse and integration on the Web." 6th International Conference on Ubiquitous Information Management and Communication. pp. 78:1–78:9 ACM, New York, NY, USA (2012).
- [CO24] Gilles, F.; Hoyer, V.; Janner, T.; Stanoevska-Slabeva, K.: "Lightweight Composition of Ad-Hoc Enterprise-Class Applications with Context-Aware Enterprise Mashups." Service-Oriented Computing. (ICSOC/ServiceWave). pp. 509–519 Springer Berlin Heidelberg, San Francisco, CA, USA (2010).
- [CO25] Govardhan, S.; Feuerlicht, G.: "Itinerary Planner: A Mashup Case Study." Nitto, E. and Ripeanu, M. 5th International Conference on Service-Oriented Computing - (ICSOC). pp. 3–14 Springer Berlin Heidelberg, Vienna, Austria (2007).
- [CO26] Gurram, R.; Mo, B.; Gueldemeister, R.: "A Web Based Mashup Platform for Enterprise 2.0." Hartmann, S., Zhou, X., and Kirchberg, M. Web Information Systems Engineering (WISE). pp. 144–151 Springer Berlin Heidelberg, Auckland, New Zealand (2008).
- [CO27] Haque, M.M.; Ahsan, A.; Yu, W.; Andersen, A.; Karlsen, R.: "Peer-to-peer orchestration of Web mashups." 14th International Conference on Information Integration and Web-based Applications & Services. pp. 294–298 ACM, New York, NY, USA (2012).
- [CO28] Hartmann, B.; Wu, L.; Collins, K.; Klemmer, S.R.: "Programming by a sample: rapidly creating Web applications with d.mix." 20th annual ACM symposium on User interface software and technology. pp. 241–250 ACM, New York, NY, USA (2007).
- [CO29] Heinzl, S.; Seiler, D.; Unterberger, M.; Nonenmacher, A.; Freisleben, B.: "MIRO: a mashup editor leveraging Web, Grid and Cloud services." 11th International Conference on Information Integration and Web-based Applications & Services. pp. 17–24 ACM, New York, NY, USA (2009).
- [CO30] Hoyer, V.; Fischer, M.: "Market Overview of Enterprise Mashup Tools." Bouguettaya, A., Krueger, I., and Margaria, T. Service-Oriented Computing (ICSOC). pp. 708–721 Springer Berlin Heidelberg, Sydney, Australia (2008).
- [CO31] Hristoskova, A.; Volckaert, B.; De Turck, F.; Dhoedt, B.: "Design of a Framework for Automated Service Mashup Creation and Execution Based on Semantic Reasoning." 5th International Conference on Internet and Web Applications and Services (ICIW). pp. 149–154, Barcelona, Spain (2010).
- [CO32] Huang, A.F.M.; Huang, S.B.; Lee, E.Y.F.; Yang, S.J.H.: "Improving End-User Programming with Situational Mashups in Web 2.0 Environment." International Symposium on Service-Oriented System Engineering (SOSE). pp. 62–67, Jhongli, Taiwan (2008).
- [CO33] Huang, G.; Zhao, Q.; Huang, J.; Liu, X.; Teng, T.; Zhang, Y.; Yuan, H.: "Towards service composition middleware embedded in Web browser." Cyber-Enabled

- Distributed Computing and Knowledge Discovery (CyberC). pp. 93–100 , Zhangjiajie, China (2009).
- [CO34] Huang, J.; Liu, X.; Zhao, Q.; Ma, J.; Huang, G.: “A browser-based framework for data cache in Web-delivered service composition.” IEEE International Conference on Service-Oriented Computing and Applications (SOCA). pp. 1–8 , Kauai, Hawaii (2010).
- [CO35] Huang, K.; Fan, Y.; Tan, W.: “An Empirical Study of Programmable Web: A Network Analysis on a Service-Mashup System.” 19th International Conference on Web Services (ICWS). pp. 552–559 , Honolulu, Hawaii, USA (2012).
- [CO36] Ikeda, S.; Nagamine, T.; Kamada, T.: “Application framework with demand-driven mashup for selective browsing.” 10th International Conference on Information Integration and Web-based Applications & Services. pp. 33–40 ACM, New York, NY, USA (2008).
- [CO37] Janiesch, C.; Fleischmann, K.; Dreiling, A.: “Extending Services Delivery with Lightweight Composition.” Hartmann, S., Zhou, X., and Kirchberg, M. Web Information Systems Engineering (WISE). pp. 162–171 Springer Berlin Heidelberg, Auckland, New Zealand (2008).
- [CO38] Janner, T.; Siebeck, R.; Schroth, C.; Hoyer, V.: “Patterns for Enterprise Mashups in B2B Collaborations to Foster Lightweight Composition and End User Development.” IEEE International Conference on Web Services (ICWS). pp. 976–983 , Los Angeles, CA, USA (2009).
- [CO39] Jung, J.; Lee, K.-H.: “Socially-Enriched Semantic Mashup of Web APIs.” Liu, C., Ludwig, H., Toumani, F., and Yu, Q. Service-Oriented Computing. pp. 389–403 Springer Berlin Heidelberg (2012).
- [CO40] Ke, W.; Wei-Jie, O.; Cheng, Z.; De-Yi, L.; Zhi-Yong, P.: “Web service schema matching based on invocation instance duplicates.” 7th International Conference on Next Generation Web Services Practices (NWeSP). pp. 386–391 , Salamanca, Spain (2011).
- [CO41] Kongdenfha, W.; Benatallah, B.; Vayssière, J.; Saint-Paul, R.; Casati, F.: “Rapid development of spreadsheet-based Web mashups.” 18th international conference on world wide Web (www). pp. 851–860 ACM, Madrid, (2009).
- [CO42] Koschmider, A.; Hoyer, V.; Giessmann, A.: “Quality metrics for mashups.” Proceedings of the 2010 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists. pp. 376–380 ACM, New York, NY, USA (2010).
- [CO43] Laga, N.; Bertin, E.; Crespi, N.: “Promoting Mashup creation through unstructured data extraction.” International Conference on Information Networking (ICOIN). pp. 59–64 , Phunket, Thailand (2012).

- [CO44] Lathem, J.; Gomadam, K.; Sheth, A.P.: "SA-REST and (S)mashups : Adding Semantics to RESTful Services." International Conference on Semantic Computing (ICSC). pp. 469–476 , Irvine, California, USA (2007).
- [CO45] Lee, E.; Joo, H.-J.: "Developing lightweight context-aware service mashup applications." 15th International Conference on Advanced Communication Technology (ICACT). pp. 1060–1064 , Pyeongchang, Korea (2013).
- [CO46] Lee, Y.-J.; Kim, J.-H.: "Semantically enabled data mashups using ontology learning method for Web APIs." Computing, Communications and Applications Conference (ComComAp). pp. 304–309 , Hong Kong, China (2012).
- [CO47] Lee, Y.-J.; Kim, J.-S.: "Automatic Web API Composition for Semantic Data Mashups." 4th International Conference on Computational Intelligence and Communication Networks (CICN). pp. 953–957 , Mathura, Uttar Pradesh, India (2012).
- [CO48] Liu, X.; Hui, Y.; Sun, W.; Liang, H.: "Towards Service Composition Based on Mashup." IEEE Congress on Services. pp. 332–339 , Salt Lake City, USA (2007).
- [CO49] Liu, X.; Jiang, N.; Zhao, Q.; Huang, G.: "iSocialMash: Convergence of social networks and services composition on a mashup framework." IEEE International Conference on Service-Oriented Computing and Applications (SOCA). pp. 1–6 , California, Irvine, USA (2011).
- [CO50] Liu, X.; Zhao, Q.; Huang, G.; Mei, H.; Teng, T.: "Composing Data-Driven Service Mashups with Tag-Based Semantic Annotations." IEEE International Conference on Web Services (ICWS). pp. 243–250 , Washington DC, USA (2011).
- [CO51] Liu, Y.; Liang, X.; Xu, L.; Staples, M.; Zhu, L.: "Using architecture integration patterns to compose enterprise mashups." European Conference on Software Architecture (WICSA/ECSA). pp. 111–120 , Sydney, Australia (2009).
- [CO52] Luo, X.; Song, M.; Liu, L.: "A solution for Mashup Platform based on SOA." Joint Conferences on Pervasive Computing (JCPC). pp. 483–488 , Tamsui, Taipei, Taiwan (2009).
- [CO53] Ma, C.; Zhang, J.; Zhao, C.; Ni, Y.; Zhang, J.; Yi, L.; Mao, X.: "Towards a More Effective Mashup Using Mashable Service Model." 6th World Congress on Services (SERVICES-1). pp. 566–573 , Miami, Florida, USA (2010).
- [CO54] Maaradji, A.; Hacid, H.; Skraba, R.; Vakali, A.: "Social Web Mashups Full Completion via Frequent Sequence Mining." IEEE World Congress on Services (SERVICES). pp. 9–16 , Wasington DC, USA (2011).
- [CO55] Makki, S.K.; Sangtani, J.: "Data Mashups & Their Applications in Enterprises." 3rd. International Conference on Internet and Web Applications and Services, (ICIW). pp. 445–450 , Athens, Greece (2008).
- [CO56] Maraïkar, Z.; Lazovik, A.; Arbab, F.: "Building Mashups for the Enterprise with SABRE." Bouguettaya, A., Krueger, I., and Margaria, T. Service-Oriented

- Computing (ICSOC). pp. 70–83 Springer Berlin Heidelberg, Sydney, Australia (2008).
- [CO57] Maximilien, E.M.; Wilkinson, H.; Desai, N.; Tai, S.: “A Domain-Specific Language for Web APIs and Services Mashups.” Krämer, B., Lin, K.-J., and Narasimhan, P. 11th International Conference on Service-Oriented Computing – (ICSOC). pp. 13–26 Springer Berlin Heidelberg, Berlin, Germany (2007).
 - [CO58] Mehandjiev, N.; Lecue, F.; Wajid, U.; Namoun, A.: “Assisted Service Composition for End Users.” 8th European Conference on Web Services (ECOWS). pp. 131–138, Ayia Napa, Cyprus (2010).
 - [CO59] Melchiori, M.: “Hybrid techniques for Web APIs recommendation.” 1st International Workshop on Linked Web Data Management. pp. 17–23 ACM, New York, NY, USA (2011).
 - [CO60] Mosser, S.; Chauvel, F.; Blay-Fornarino, M.; Riveill, M.: “Web Services Composition: Mashups Driven Orchestration Definition.” International Conference on Computational Intelligence for Modelling Control Automation. pp. 284–289, Vienna, Austria (2008).
 - [CO61] Ngu, A.H.H.; Carlson, M.P.; Sheng, Q.Z.; Paik, H.: “Semantic-Based Mashup of Composite Applications” ; IEEE Transactions on Services Computing, 3, 1, 2–15 (2010).
 - [CO62] Pérez, S.; Díaz, O.: “Mashup-Aware Corporate Portals.” Chen, L., Triantafyllou, P., and Suel, T. Web Information Systems Engineering (WISE). pp. 271–278 Springer Berlin Heidelberg, Hong Kong, China (2010).
 - [CO63] Pietschmann, S.; Nestler, T.; Daniel, F.: “Application composition at the presentation layer: alternatives and open issues.” 12th International Conference on Information Integration and Web-based Applications and Services. pp. 461–468 ACM, New York, NY, USA (2010).
 - [CO64] Pietschmann, S.; Radeck, C.; Meißner, K.: “Semantics-based discovery, selection and mediation for presentation-oriented mashups.” 5th International Workshop on Web APIs and Service Mashups. pp. 7:1–7:8 ACM, New York, NY, USA (2011).
 - [CO65] Pietschmann, S.; Waltsgott, J.; Meißner, K.: “A Thin-Server Runtime Platform for Composite Web Applications.” 5th International Conference on Internet and Web Applications and Services (ICIW). pp. 390–395, Barcelona, Spain (2010).
 - [CO66] Riabov, A. V.; Boillet, E.; Feblowitz, M.D.; Liu, Z.; Ranganathan, A.: “Wishful search: interactive composition of data mashups.” Proceedings of the 17th international conference on World Wide Web. pp. 775–784 ACM, New York, NY, USA (2008).
 - [CO67] Ro, A.; Xia, L.-Y.; Paik, H.-Y.; Chon, C.: “Bill Organiser Portal: A Case Study on End-User Composition.” Hartmann, S., Zhou, X., and Kirchberg, M. Web Information

Systems Engineering (WISE). pp. 152–161 Springer Berlin Heidelberg, Auckland, New Zealand (2008).

- [CO68] Roy Chowdhury, S.; Daniel, F.; Casati, F.: “Efficient, Interactive Recommendation of Mashup Composition Knowledge.” Kappel, G., Maamar, Z., and Motahari-Nezhad, H. 9th international conference on Service-Oriented Computing (ICSOC). pp. 374–388 Springer Berlin Heidelberg, Berlin, Germany (2011).
- [CO69] Sabbouh, M.; Higginson, J.L.; Wan, C.; Bennett, S.R.: “Using Mapping Relations to Semi Automatically Compose Web Services.” Congress on Services - Part I. IEEE. pp. 211–218, Salt Lake City, USA (2008).
- [CO70] Dos Santos, C.R.P.; Bezerra, R.S.; Ceron, J.M.; Granville, L.Z.; Rockenbach Tarouco, L.M.: “On using mashups for composing network management applications” ; Communications Magazine, IEEE, 48, 12, 112–122 (2010).
- [CO71] Dos Santos, C.R.P.; Bezerra, R.S.; Granville, L.Z.; Bertholdo, L.M.; Cheng, W.; Anerousis, N.: “A data confidentiality architecture for developing management mashups.” International Symposium on Integrated Network Management (IM-IFIP/IEEE). pp. 49–56, Dublin, Ireland (2011).
- [CO72] Soylu, A.; Wild, F.; Mödritscher, F.; Desmet, P.; Verlinde, S.; De Causmaecker, P.: “Mashups and widget orchestration.” Proceedings of the International Conference on Management of Emergent Digital EcoSystems. pp. 226–234 ACM, New York, NY, USA (2011).
- [CO73] Stecca, M.; Fornasa, M.; Dall’Armellina, N.; Maresca, M.: “Event-Driven Mashup Orchestration with Scala.” 9th International Conference on Services Computing (SCC),. pp. 531–538, Honolulu, Hawaii, USA (2012).
- [CO74] Stecca, M.; Maresca, M.: “An Architecture for a Mashup Container in Virtualized Environments.” 3rd International Conference on Cloud Computing (CLOUD). pp. 386–393, Miami, FL, USA (2010).
- [CO75] Stecca, M.; Maresca, M.: “An execution platform for event driven mashups.” 11th International Conference on Information Integration and Web-based Applications & Services. pp. 33–40 ACM, Kuala Lumpur, Malaysia (2009).
- [CO76] Stecca, M.; Maresca, M.: “Mashup Patterns from Service Component Taxonomy.” 24th International Conference on Advanced Information Networking and Applications Workshops (WAINA). pp. 12–17, Bradford, United Kingdom (2010).
- [CO77] Stolee, K.T.; Elbaum, S.: “Refactoring pipe-like mashups for end-user programmers.” 33rd International Conference on Software Engineering (ICSE),. pp. 81–90, Waikiki, Honolulu, Hawaii (2011).
- [CO78] Torres, V.; Pérez, J.M.; Koschmider, A.; Daniel, F.: “Dealing with collaborative tasks in process mashups.” 5th International Workshop on Web APIs and Service Mashups. pp. 4:1–4:8 ACM, New York, NY, USA (2011).

- [CO79] Treiber, M.; Kritikos, K.; Schall, D.; Dustdar, S.; Plexousakis, D.: "Modeling context-aware and socially-enriched mashups." 3rd and 4th International Workshop on Web APIs and Services Mashups. pp. 2:1–2:8 ACM, New York, NY, USA (2010).
- [CO80] Tritschler, M.; Kleinfeld, R.; Steglich, S.: "Enhancing Interoperability in Cross-Platform Enterprise Mashups through Data Aggregation and Extraction." 13th International Conference on High Performance Computing and Communications (HPCC). pp. 695–700 , Banff, AB, Canadá (2011).
- [CO81] Väänänen-Vainio-Mattila, K.; Wäljas, M.: "Towards user-centered mashups: exploring user needs for composite Web services." Extended Abstracts on Human Factors in Computing Systems (CHI). pp. 1327–1332 ACM, New York, NY, USA (2011).
- [CO82] Vallejos, J.; Huang, J.; Costanza, P.; De Meuter, W.; D'Hondt, T.: "A programming language approach for context-aware mashups." 3th and 4th International Workshop on Web APIs and Services Mashups. pp. 4:1–4:5 ACM, New York, NY, USA (2010).
- [CO83] Vancea, A.; Grossniklaus, M.; Norrie, M.C.: "Database-Driven Web Mashups." 8th International Conference on Web Engineering (ICWE). pp. 162–174 , Yorktown Heights, New York, USA (2008).
- [CO84] Wang, G.; Liu, J.; Cao, B.; Tang, M.: "Mashup Service Classification and Recommendation Based on Similarity Computing." 2nd International Conference on Cloud and Green Computing (CGC). pp. 621–628 , Xiangtan, Hunan, China (2012).
- [CO85] Wang, G.; Yang, S.; Han, Y.: "Mashroom: end-user mashup programming using nested tables." 18th Int. World Wide Web Conference (WWW). pp. 861–870 , Madrid, Spain (2009).
- [CO86] Weiss, M.; Sari, S.: "Evolution of the mashup ecosystem by copying." 3rd and 4th International Workshop on Web APIs and Services Mashups. pp. 11:1–11:7 ACM, New York, NY, USA (2010).
- [CO87] Westerski, A.: "Integrated Environment for Visual Data-Level Mashup Development." Vossen, G., Long, D.E., and Yu, J. Web Information Systems Engineering (WISE). pp. 481–487 Springer Berlin Heidelberg, Poznan, Poland (2009).
- [CO88] Wohlstadtter, E.; Li, P.; Cannon, B.: "Web Service Mashup Middleware with Partitioning of XML Pipelines." IEEE International Conference on Web Services (ICWS). pp. 91–98 , Los Angeles, CA, USA (2009).
- [CO89] Xie, L.; Xu, L.; de Vrieze, P.: "Process modelling in process-oriented enterprise mashups." 2nd IEEE International Conference on Information Management and Engineering (ICIME). pp. 650–654 , Cape Town, South Africa (2010).

- [CO90] Ye, W.; Luo, R.; Zhang, S.: "A Process Mashup Model Based on Complex Event Processing." 7th International Symposium on Service-Oriented System Engineering. pp. 181–185 IEEE Computer Society, Washington, DC, USA (2013).
- [CO91] Ye, W.; Luo, R.; Zhang, S.; Zhong, M.: "A Lightweight Process Mashup Approach Based on Business Unit." 7th International Symposium on Service-Oriented System Engineering. pp. 203–211 IEEE Computer Society, Washington, DC, USA (2013).
- [CO92] Yu, J.; Benatallah, B.; Casati, F.; Daniel, F.: "Understanding Mashup Development." IEEE Internet Computing. pp. 44–52 IEEE Educational Activities Department, Piscataway, NJ, USA (2008).
- [CO93] Zhang, X.; He, K.; Wang, J.; Liu, J.; Wang, C.; Lu, H.: "On-Demand Service-Oriented MDA Approach for SaaS and Enterprise Mashup Application Development." International Conference on Cloud and Service Computing (CSC). pp. 96–103 , Shanghai, China (2012).
- [CO94] Zhao, Q.; Huang, G.; Huang, J.; Liu, X.; Mei, H.: "A Web-Based Mashup Environment for On-the-Fly Service Composition." IEEE International Symposium on Service-Oriented System Engineering (SOSE). pp. 32–37 , San Francisco Bay, USA (2008).
- [CO95] Zhao, Q.; Liu, X.; Huang, G.; Huang, J.; Mei, H.: "A Browser-Based Middleware for Service-Oriented Rich Client." International Conference on Service Sciences (ICSS). pp. 22–27 , Rome, Italy (2010).
- [CO96] Zhao, Z.; Bhattarai, S.; Liu, J.; Crespi, N.: "Mashup services to daily activities: end-user perspective in designing a consumer mashups." 13th International Conference on Information Integration and Web-based Applications and Services. pp. 222–229 ACM, New York, NY, USA (2011).
- [CO97] Zhou, C.; Chen, H.; Peng, Z.; Ni, Y.; Xie, G.: "Ontology-driven mashup auto-completion on a data API network" ; Tsinghua Science and Technology, 15, 6, 657–667 (2010).

APENDICE C - PLANTILLAS

Documento de Especificación de Requisitos de Evaluación

a) Estado en el que se Ejecutará la Evaluación de Calidad (Marcar con una X)

-Evaluación formativa en la selección de componentes	
-Evaluación formativa en la composición	
-Evaluación sumativa del producto terminado	
-Evaluación a lo largo de todos los estados del ciclo de vida	

b) Requisitos de la Calidad del Mashup

-Stackeholders involucrados

-Usuario	
-Compositor	
-Otros (Especificar)	

c) Lista de Atributos a Evaluar

No.	Atributo	Código Atributo
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		
...		

APENDICE D - PLANTILLAS DE CASOS DE ESTUDIO

Documento de Especificación de Requisitos de Evaluación

a) Estado en el que se Ejecutará la Evaluación de la Usabilidad (Marcar con una X)

-Evaluación formativa en la selección de componentes	
-Evaluación formativa en la composición	
-Evaluación sumativa del producto terminado	x
-Evaluación a lo largo de todos los estados del ciclo de vida	

b) Requisitos de la Usabilidad del Mashup

-Stakeholders involucrados

-Usuario	x
-Compositor	x
-Otros (Especificar)	

c) Lista de Atributos a Evaluar

No.	Atributo	Código Atributo
1	Disposición de los componentes	1.1.4
2	Tamaño del componente	1.1.5
3	Coherencia en la agrupación de componentes	1.2.1
4	Densidad de la información	1.2.2
5	Popularidad de los componentes	1.3.4
6	Presentación de controles en los componentes	1.3.5
7	Actualización de información entre componentes	1.4.4
8	Propósito fácilmente distinguible de los componentes	2.1.4
9	Distinción de posibles acciones de los componentes	2.2.3
10	Facilidad para alcanzar los componentes	3.2.8
11	Carga de los componentes	3.2.9
12	Comportamiento correcto de los componentes	3.4.6
13	Independencia de dispositivo	3.5.1
14	Auto-ajuste de los componentes en los dispositivos	3.5.2
15		
...		

Documento de Especificación de Requisitos de Evaluación

a) Estado en el que se Ejecutará la Evaluación de la Usabilidad (Marcar con una X)

-Evaluación formativa en la selección de componentes	
-Evaluación formativa en la composición	
-Evaluación sumativa del producto terminado	X
-Evaluación a lo largo de todos los estados del ciclo de vida	

b) Requisitos de la Usabilidad del Mashup

-Stakeholders involucrados

-Usuario	x
-Compositor	x
-Otros (Especificar)	
Evaluador	

c) Lista de Atributos a Evaluar

No.	Atributo	Código Atributo
1	Disposición de los componentes	1.1.4
2	Tamaño del componente	1.1.5
3	Coherencia en la agrupación de componentes	1.2.1
4	Popularidad de los componentes	1.3.4
5	Actualización de información entre componentes	1.4.4
6	Propósito fácilmente distinguible de los componentes	2.1.4
7		
8		
9		
10		
11		
12		
13		
14		
15		
...		

Documento de Especificación de Requisitos de Evaluación

a) Estado en el que se Ejecutará la Evaluación de la Usabilidad (Marcar con una X)

-Evaluación formativa en la selección de componentes	
-Evaluación formativa en la composición	
-Evaluación sumativa del producto terminado	X
-Evaluación a lo largo de todos los estados del ciclo de vida	

b) Requisitos de la Usabilidad del Mashup

-Stakeholders involucrados

-Usuario	x
-Compositor	x
-Otros (Especificar)	
Evaluador	

c) Lista de Atributos a Evaluar

No.	Atributo	Código Atributo
1	Densidad de la información	1.2.2
2	Popularidad de los componentes	1.3.4
3	Presentación de controles en los componentes	1.3.5
4	Actualización de información entre componentes	1.4.4
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		
...		